

設計與實作路燈管理系統：配置與佈署

Design and Implementation of a Street Light

Management System: Configuration and Deployment

REPORTER: GUAN-SHENG CHEN

ADVISOR: PROF. CHU-SING YANG

DATE: 2019/07/15

Outline

- Introduction
- Background and Related Works
- System Design
- Experiment and Verification
- Conclusion and Future Works

Outline

- **Introduction**
- Background and Related Works
- System Design
- Experiment and Verification
- Conclusion and Future Works

Introduction

- Through integrating various **sensors** and **embedded devices** provides the opportunity to transform **streetlights** into a multifunctional digital urban platform.
- However, existing systems are far less **flexible** and developed without **security and privacy** protections.
- Our work implements a streetlight management system which is
 - **Highly scalable**
 - modular, rapid deployment, high resource utilization
 - **Security oriented**
 - transmit data with authentication (permission) and encryption
 - **User friendly**
 - easy-to-deploy, real-time feedback, multiuser accessibility

Outline

- Introduction
- Background and Related works
- System Design
- Experiment and Verification
- Conclusion and Future works

Related Works

- Re-Imagining Streetlight Infrastructure as a Digital Urban Platform ... Journal of Urban Technology, 2017
- Smart IP – **central management system** for public lighting in Portugal ... CIRED Journal, 2017
- Design of Smart LED Streetlight System for Smart City With **Web-Based Management System** ... IEEE Sensors Journal, 2017
- Smart Outdoor Light Desktop **Central Management System** ... IEEE Intelligent Transportation Systems Magazine, 2018
- SSL: Smart Street Lamp Based on Fog Computing for Smarter Cities ... IEEE Transactions on Industrial Informatics, 2018
- Our proposed system pays more attention to
 - Manageability and Scalability of **deployment**
 - Encryption and Authentication for **protecting communications**

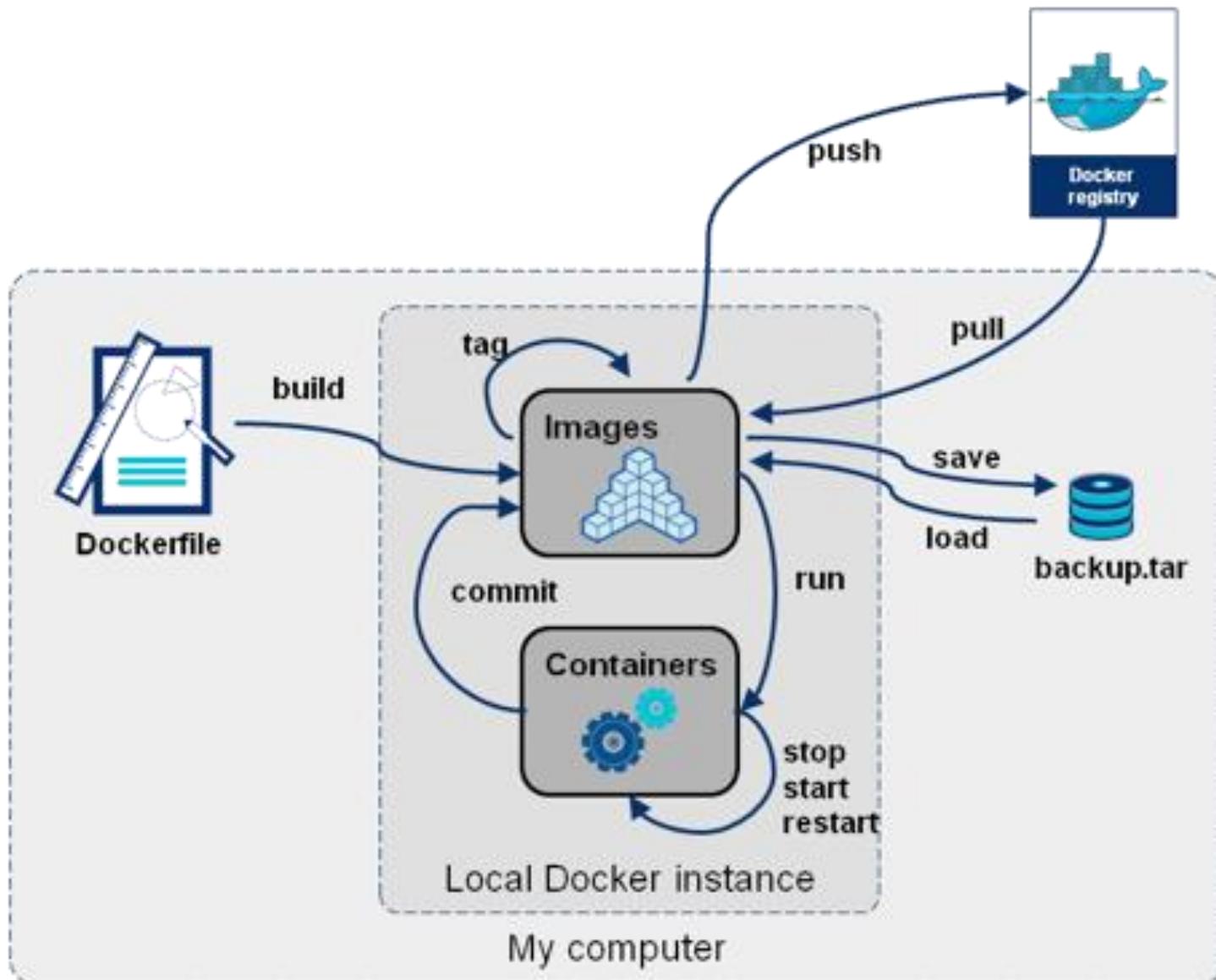
Container vs. Virtual Machine

- Docker is a high-level containerization API

Table 2.1: Comparison of Docker and Virtual Machines [2]

Feature	Docker	Virtual Machines
Start time	<50 ms	30 - 45 seconds
Stop time	<50 ms	5 - 10 seconds
System Overhead	No overhead	Overhead due to hypervisor
Storage space	Tens of MBs in size	Tens of GBs in size
Scalability	Highly scalable	Not easily scalable
CPU load when idle	Normal	~1.5% more than docker
Isolation	Less isolation due to software virtualization	More isolation due to hardware virtualization
Network round trip latency	~75 μ s	~60 μ s
I/O throughput (Read and Write)	100000 I/Os per second	50000 I/Os per second

Docker Architecture



Containerization in IoT environment

- Feasibility of Fog Computing Deployment based on Docker **Containerization** over RaspberryPi ... ICDCN, ACM, 2017
- Virtualization on Internet of Things **Edge** Devices With **Container** Technologies: A Performance Evaluation ... IEEE Access, 2017
- **Container-as-a-Service at the Edge**: Trade-off between Energy Efficiency and Service Availability at Fog Nano Data Centers ... IEEE Wireless Communications, 2017
- **Edge** Computing Embedded Platform with **Container** Migration ... IEEE SmartWorld, 2017
- Evaluating Performance of **Containerized** IoT Services for Clustered Devices at the Network **Edge** ... IEEE Internet of Things Journal, 2017
- VIoLET: A Large-scale **Virtual** Environment for **Internet of Things** ... CoRR, arXiv, 2018
- Video Processing on the **Edge** for Multimedia IoT Systems ... CoRR, arXiv, 2018

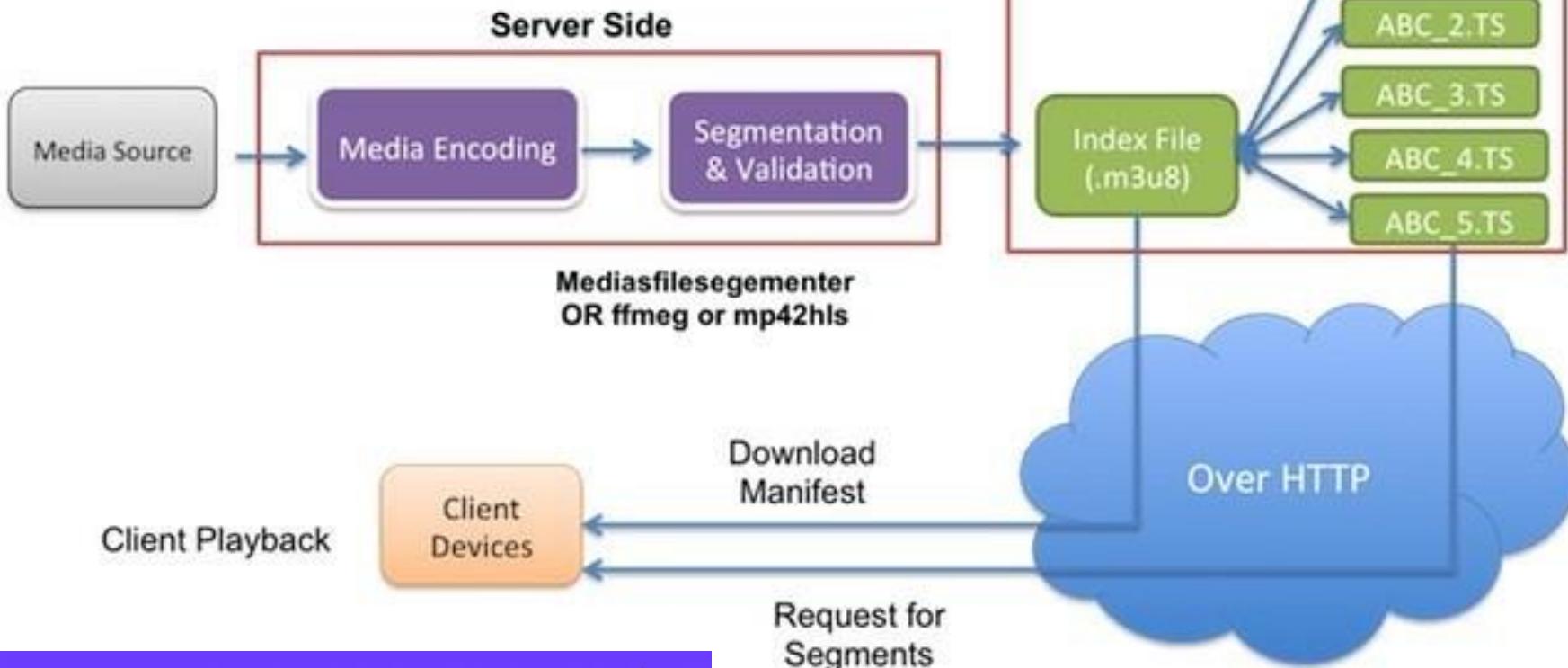
Live Streaming Technology

- Streaming protocols
 - RTSP: IP camera
 - RTMP: Encode and publish to streaming platform
- HTTP-based streaming protocols

	HLS	MPEG-DASH	HTTP-FLV
Open Source Client	hls.js	dash.js	flv.js
Open Source Server	nginx-rtmp-module	nginx-rtmp-module	nginx-http-flv-module
Support iOS Safari	Yes (natively)	No	No
Example	Twitch, LINE TV	Netflix, YouTube Live	Douyu TV, bilibili

HTTP Live Streaming

HLS Streaming Workflow

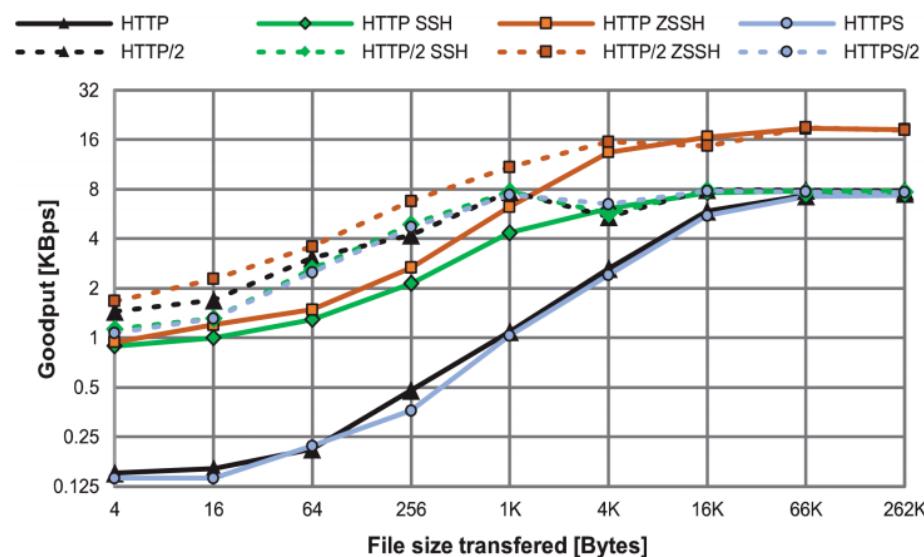


Basic HLS Encryption

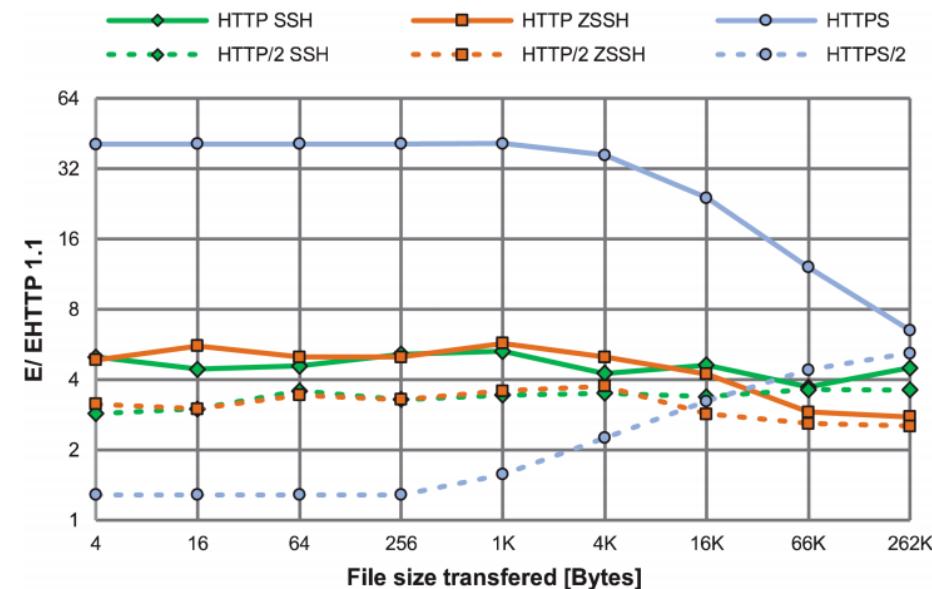


SSH Tunnel in IoT

- IoTsafe, Decoupling Security From Applications for a Safer IoT ... IEEE Access, 2019
- HTTP over TLS vs. HTTP over SSH tunnel (with ZLIB compression)
 - performed over 802.15.4 LoWPAN on top of two Raspberry Pis



(a) Goodput

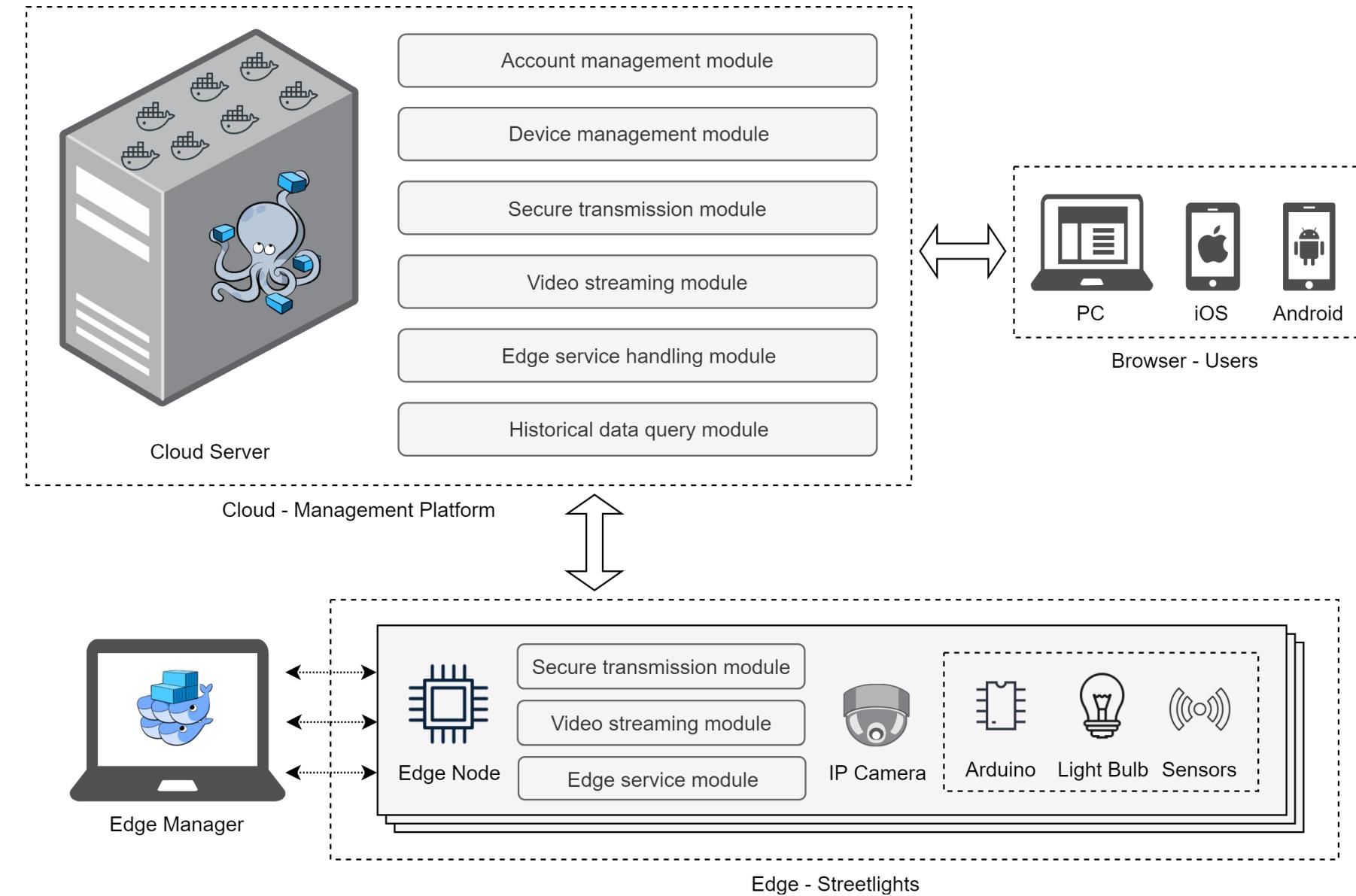


(b) CPU energy consumption

Outline

- Introduction
- Background and Related works
- **System Design**
- Experiment and Verification
- Conclusion and Future works

System Architecture



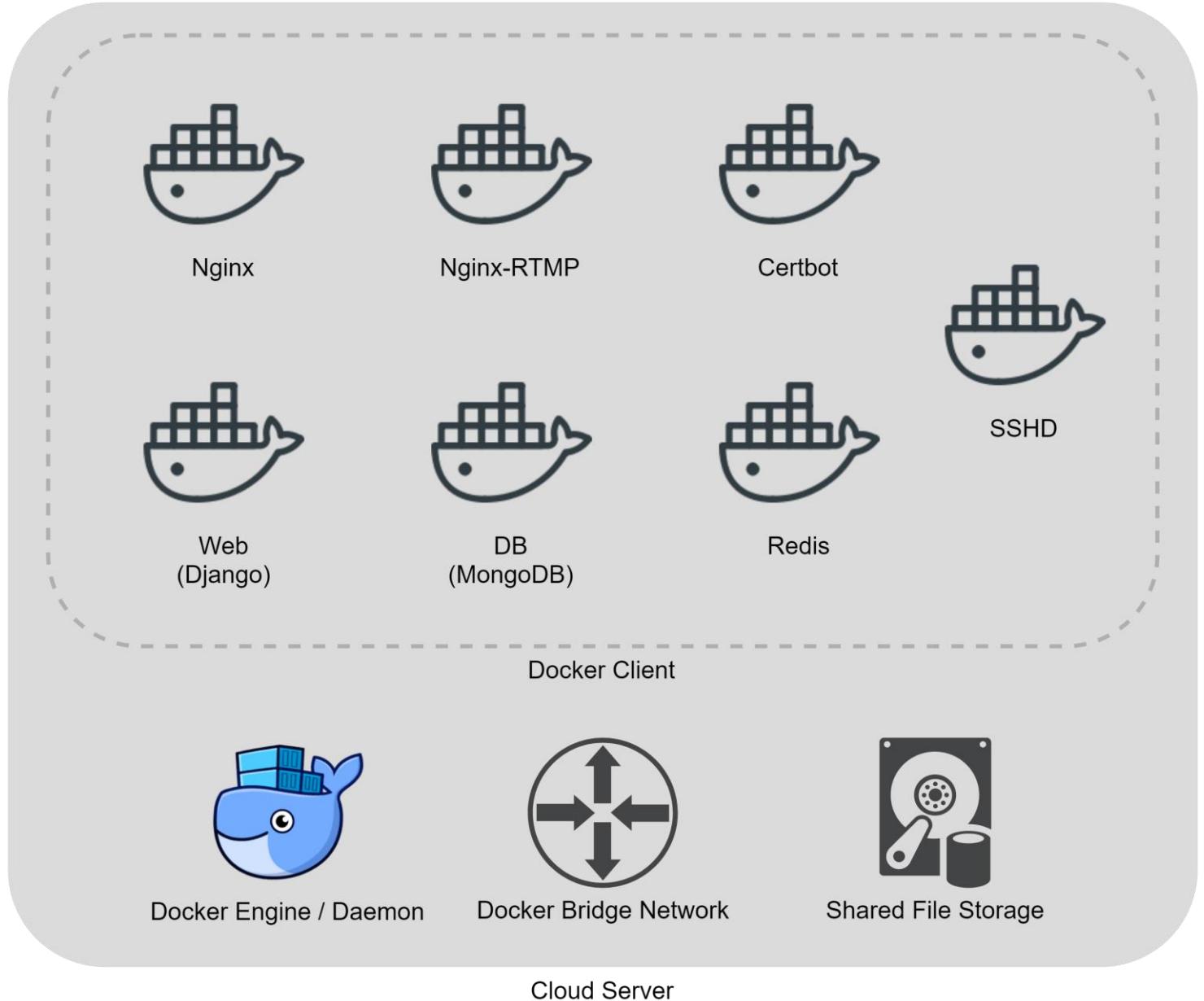
Outline

- Introduction
- Background and Related works
- System Design
 - Cloud Management System
 - Edge Services Orchestrator
- Experiment and Verification
- Conclusion and Future works

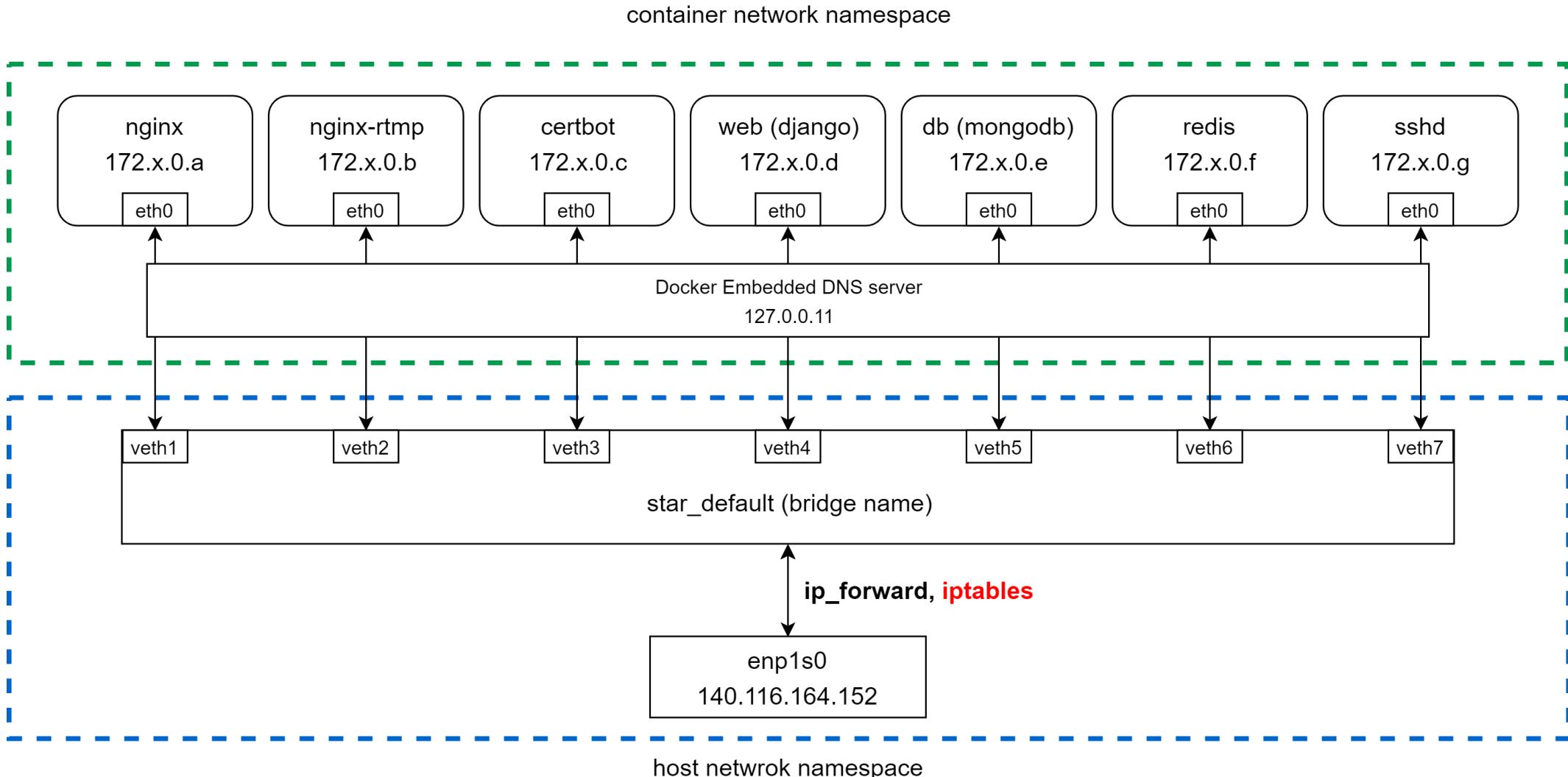
Cloud Management Platform

- **Cloud Server Architecture**
 - Overview
 - Network
 - Docker images, containers
- **Cloud Server Modules**
 - Account management
 - Device management
 - Secure transmission
 - Video streaming
 - Edge service handling
 - Historical data query

Cloud Server - Overview



Cloud Server – Docker Bridge Network



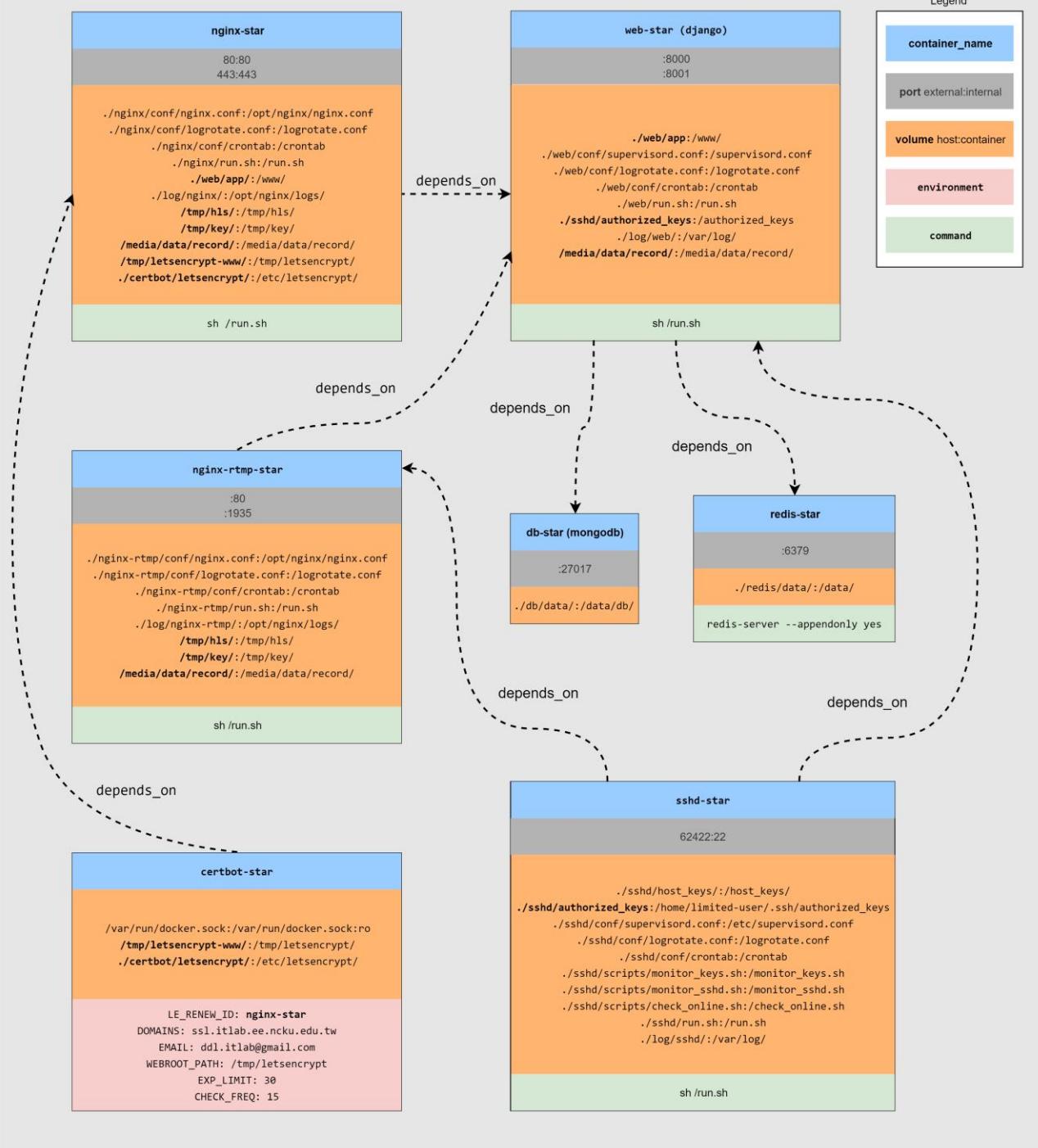
Cloud Server - Docker Images

Image	Info	Base Image	Size
Nginx		alpine:3.9	23MB
Nginx-RTMP		alpine:3.9	112MB
Web (Django)		python:3.7.3-alpine3.9	280MB
MongoDB		mvertes/alpine-mongo:4.0.61	123MB
Redis		redis:5.0.5-alpine3.9	50.9MB
SSHD		alpine:3.9	57MB
Certbot		docker:18.09.6	256MB

Cloud Server - Docker Containers

Ports exposed:

- 80
(HTTPS certificate renewal)
- 443
(HTTPS requests)
- 62422
(SSH Tunnel)



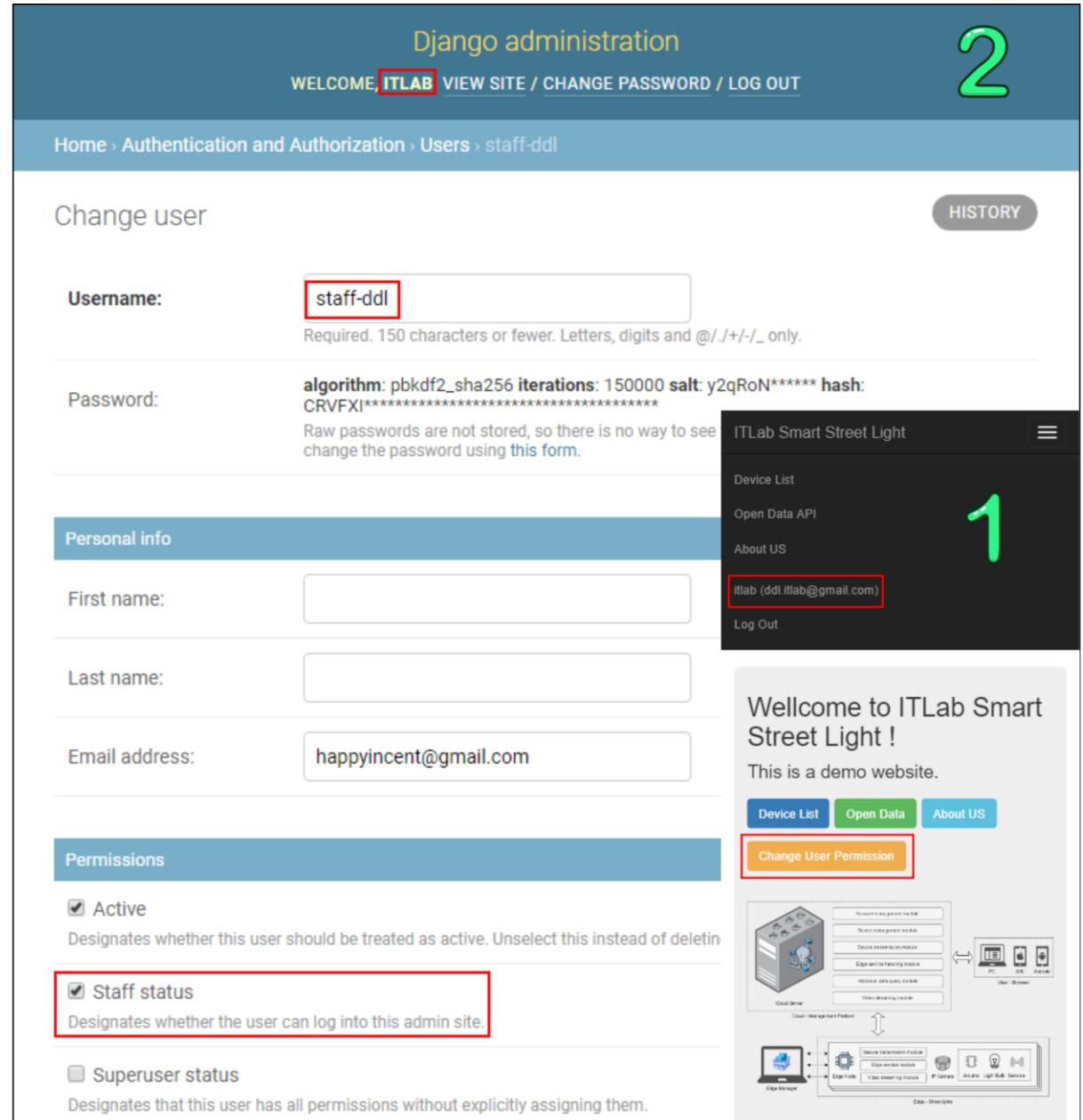
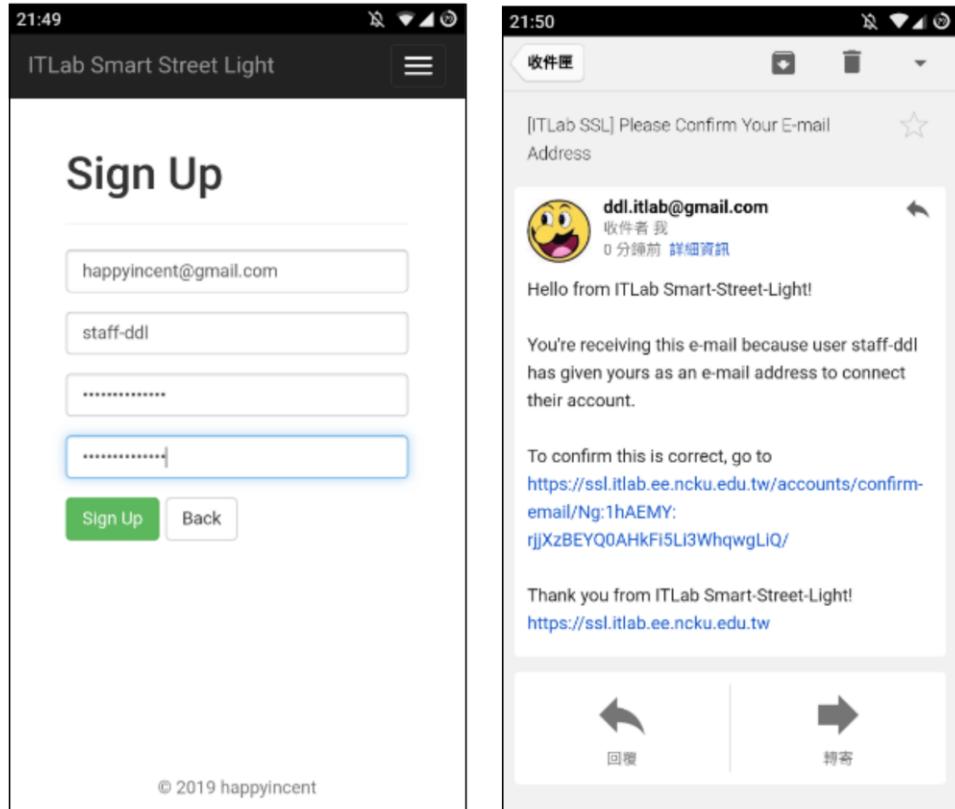
Cloud Management Platform

- Cloud Server Architecture
 - Overview
 - Network
 - Docker images, containers
- **Cloud Server Modules**
 - Account management
 - Device management
 - Secure transmission
 - Video streaming
 - Edge service handling
 - Historical data query

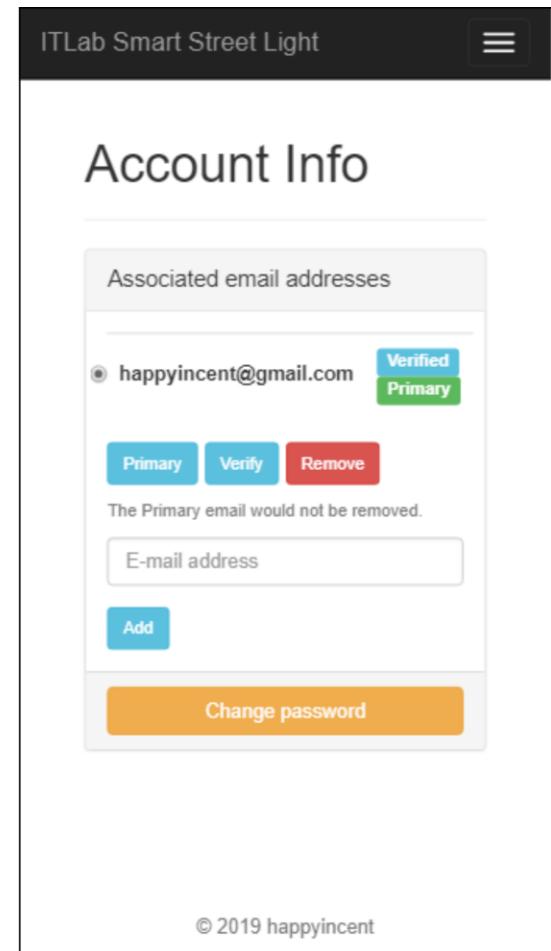
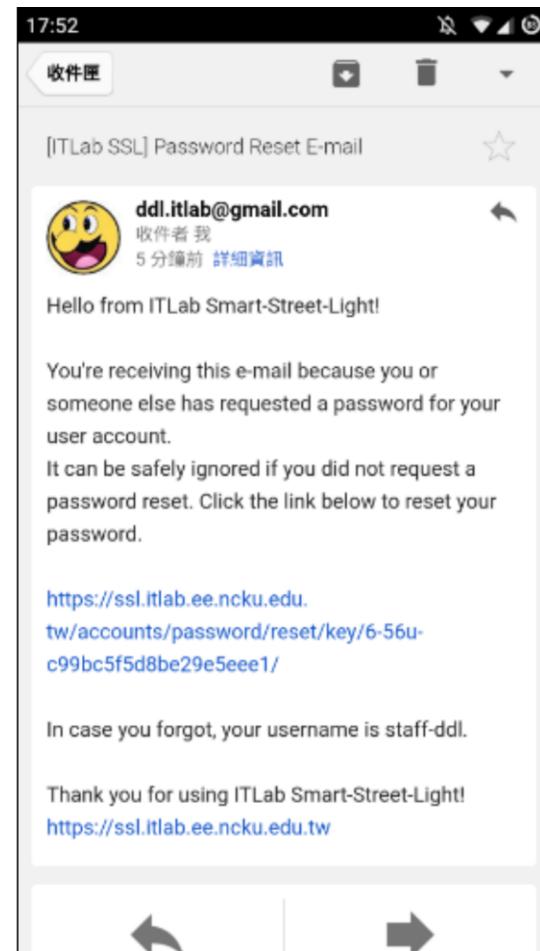
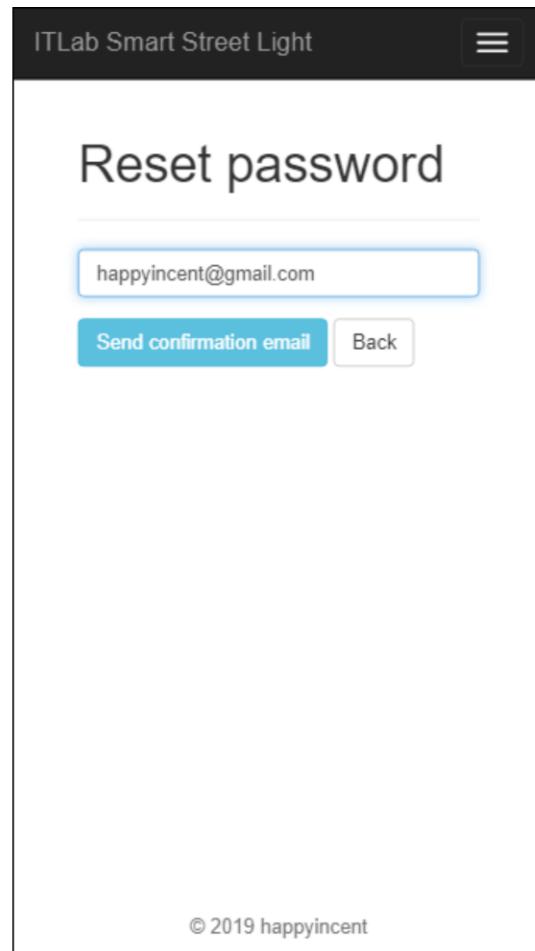
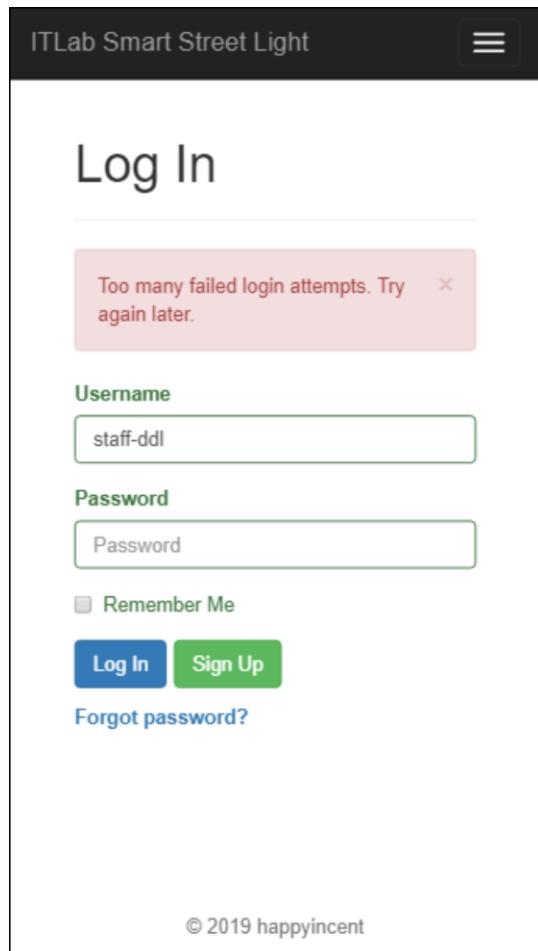
Account Management

Type \ Permission	Get Data/Streaming	Config LED (Timetable)	Setup Device Configuration	Edit User Permission
Superuser	✓	✓	✓	✓
Staff	✓	✓	✓ (creator)	
Email Verified	✓			
Email not Verified				

Account Management



Account Management



Device Management

- Maintain Device Collection (stored in MongoDB)
 - User ID
 - Device ID
 - Longitude / Latitude
 - Token
 - SSH Public Key
 - LED / PIR schedule
- Process of CRUD (with permission check)
 - Generate and store token into Redis
(create device, reset token)
 - Drop connections
(update public key, reset token, delete device)

Device Management

The image displays five screens related to device management:

- New Device:** A form to add a new device. It includes fields for Device ID (unmodifiable, set to "TMP"), Longitude (120.22283), Latitude (22.99472), and SSH Public Key (ssh-rsa [key] [comment]). Buttons for Save and Back are at the bottom.
- Edit Device TMP:** A form to edit an existing device named "TMP". It shows the current values for Longitude (120.22283), Latitude (22.99472), and SSH Public Key (ssh-rsa [key] [comment]). It also includes a Token field (1fc047d1-7adf-4c27-9ce4-8ce3) with a Reset button, and a PIR Timeout (ms) section with a dropdown menu showing "Waiting Edge ..." and an Update button. Buttons for Save and Back are at the bottom.
- ERROR:** An error message stating "Fail to update Device \"TMP\"". It advises to "Make sure the **SSH Public Key** format is correct." A Back button is present.
- Delete Device TMP:** A confirmation dialog asking "Are you sure you want to delete TMP ?" with "Confirm" and "Back" buttons.
- Mobile Browser Screenshot:** A screenshot of a mobile browser window titled "Edit Device | ITLab SS". The URL is https://ssl.itlab.ee.ncku.edu.tw/device_update/VM0_1. The page shows a 403 Forbidden error. The browser status bar indicates the time is 17:24 and shows LTE signal strength.

© 2019 happyincent

Device Management

The diagram illustrates the ITLab Smart Street Light system architecture. At the top, a navigation bar includes a logo, the text "ITLab Smart Street Light", and a menu icon. Below the navigation bar is a "Device List" button. The main content area features a large welcome message: "Wellcome to ITLab Smart Street Light ! This is a demo website." Below this message are three buttons: "Device List" (blue), "Open Data" (green), and "About US" (light blue). A red box highlights the "Device List" button. To the left of the main content is a "Cloud Managed Platform" section containing a server icon and a list of management modules: Account management module, Device management module, Data transmission module, Cloud service Pending module, Historical data query module, and Video monitoring module. An arrow points from this section to a "User - Resource" section on the right, which lists PC, iOS, and Android devices.

ITLab Smart Street Light			
ID	Staff	Info	Edit
TX2_1	staff-ddl	Now History	Config Delete
RPi_1	Q36064206	Now History	Config Delete
VM0_1	itlab	Now History	Config Delete
VM0_2	itlab	Now History	Config Delete

ITLab Smart Street Light

≡

Edit Device TX2_1

Longitude

Latitude

SSH Public Key

Token

 Reset

PIR Timeout (ms)

 Update

Save Back

The figure shows a mobile application interface for managing IoT devices. At the top, there are two entries for 'itlab' devices:

- VM0_17: Buttons for 'Now', 'Config', 'History', and 'Delete'.
- VM0_18: Buttons for 'Now', 'Config', 'History', and 'Delete'.

Below these are two large buttons: 'Add New Device' (blue) and 'Deployment Profiles' (green).

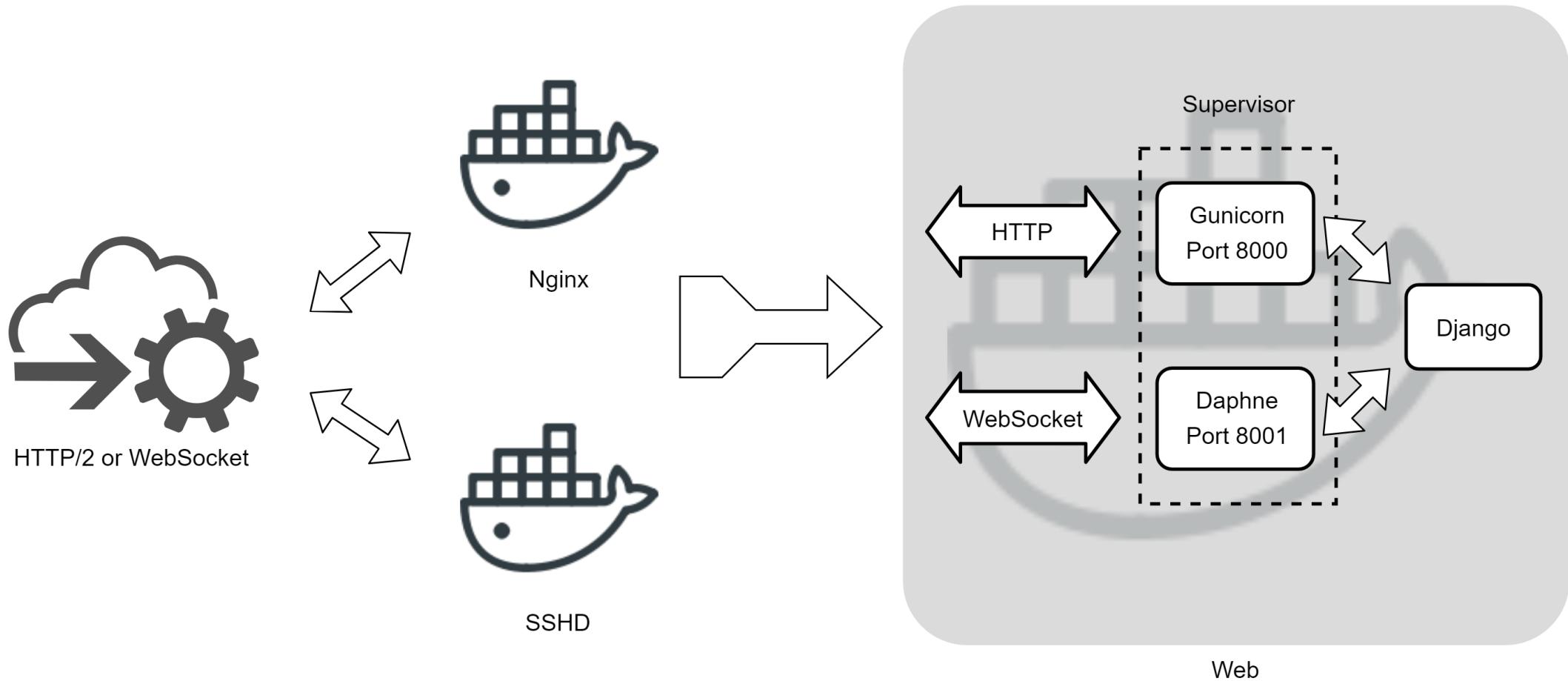
The main area is a map of a university campus. A legend on the left shows a plus sign for adding devices and a minus sign for removing them. Two devices are currently deployed on the map:

- RPi_1**: Located near the center of the map, marked with a blue icon.
- TX2_1**: Located further west, marked with a blue icon.

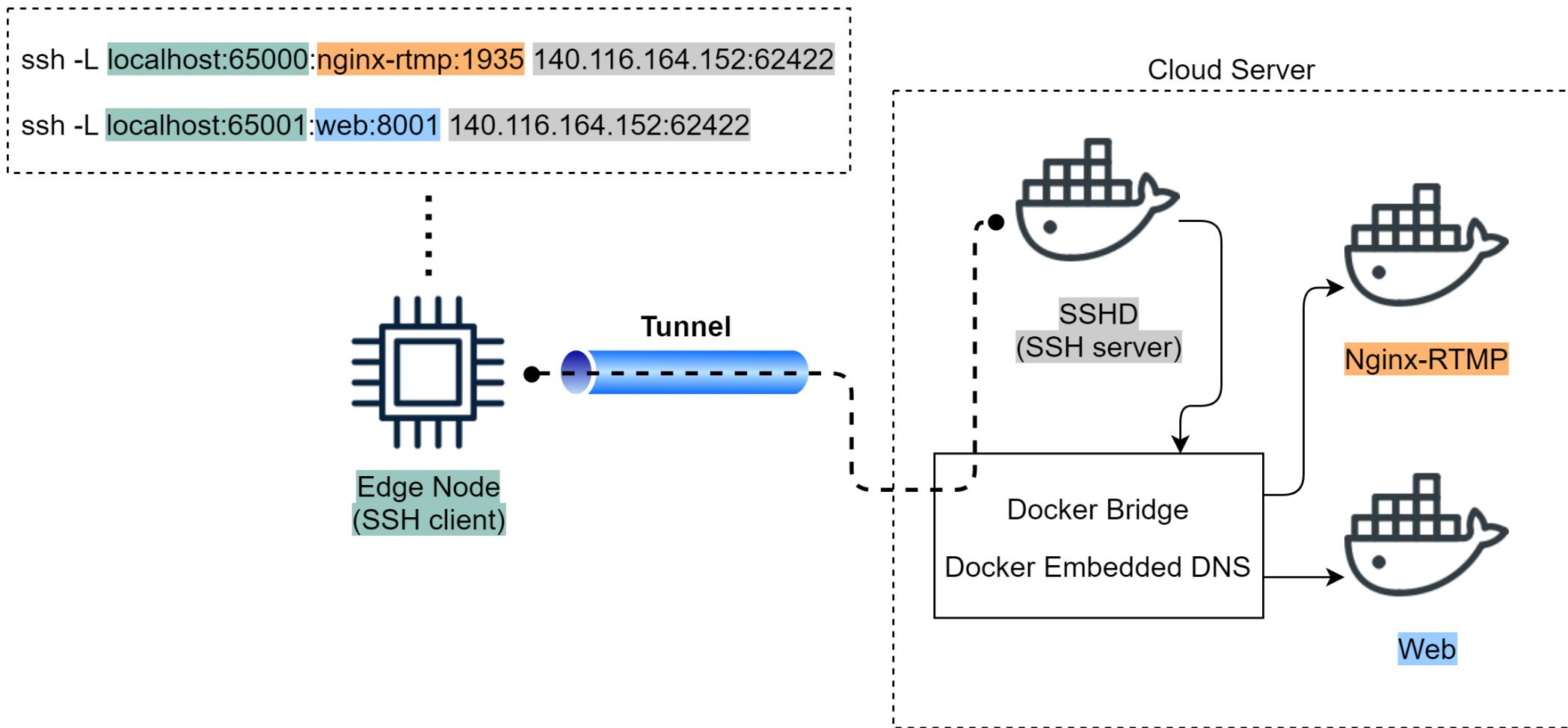
The map includes several roads and landmarks labeled in Chinese, such as '小東路', '成功大學', '來豐路', '東和路', '東寧路', '經義堂', '崇基', '中正堂', '東寧運動公園', and '崇基'. There are also green areas representing parks or fields.

At the bottom of the screen, there is a footer with the text: 'Leaflet | Map data © OpenStreetMap contributors, CC-BY-SA, Imagery © Mapbox'.

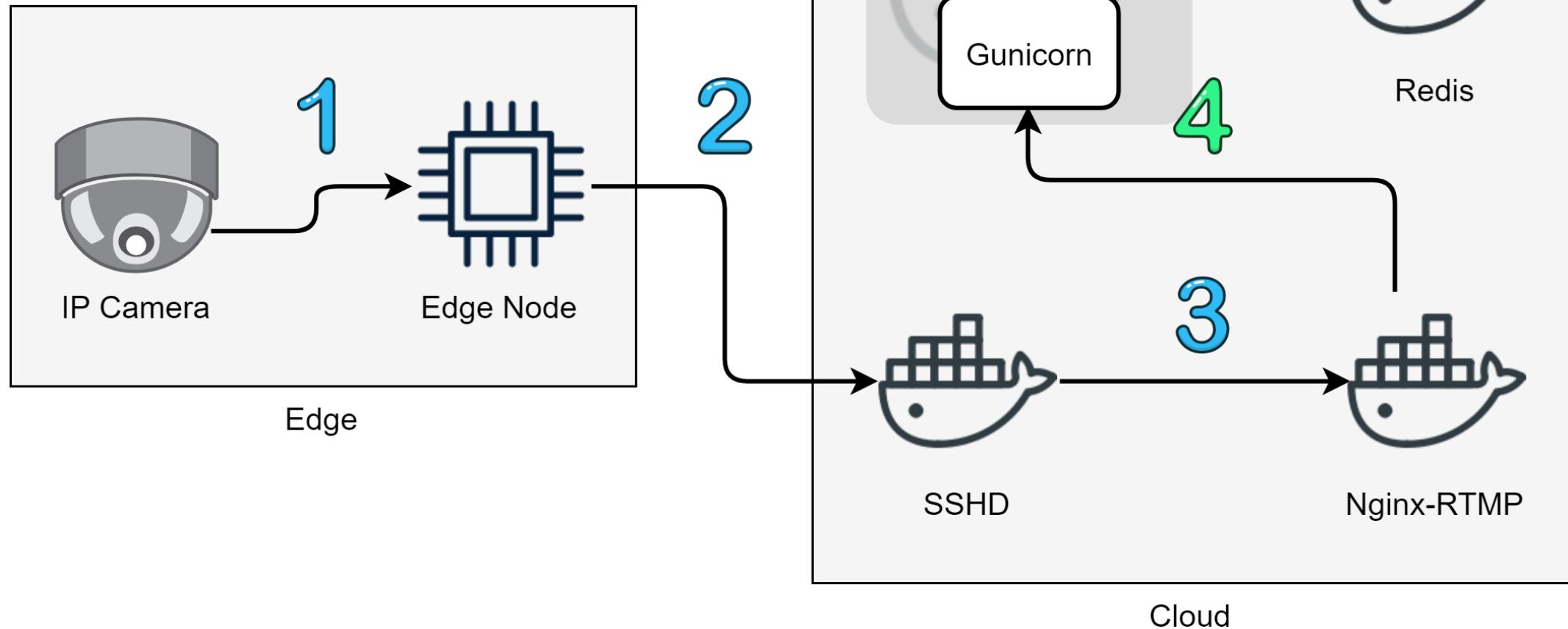
Web-based Requests Workflow



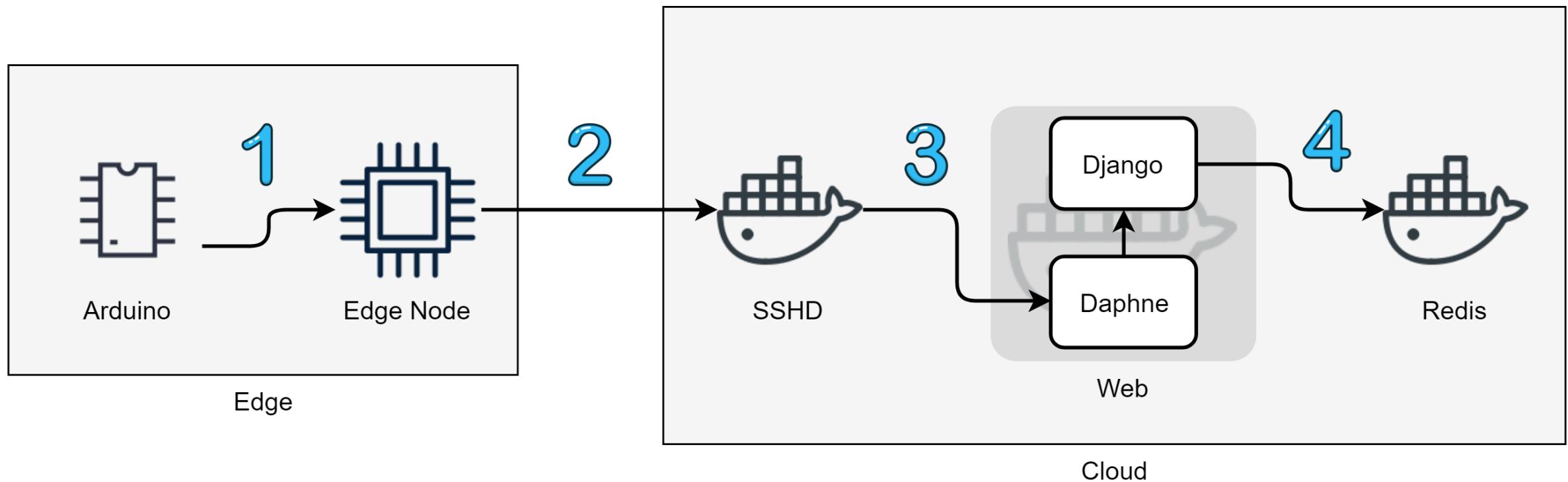
Secure Transmission (Edge)



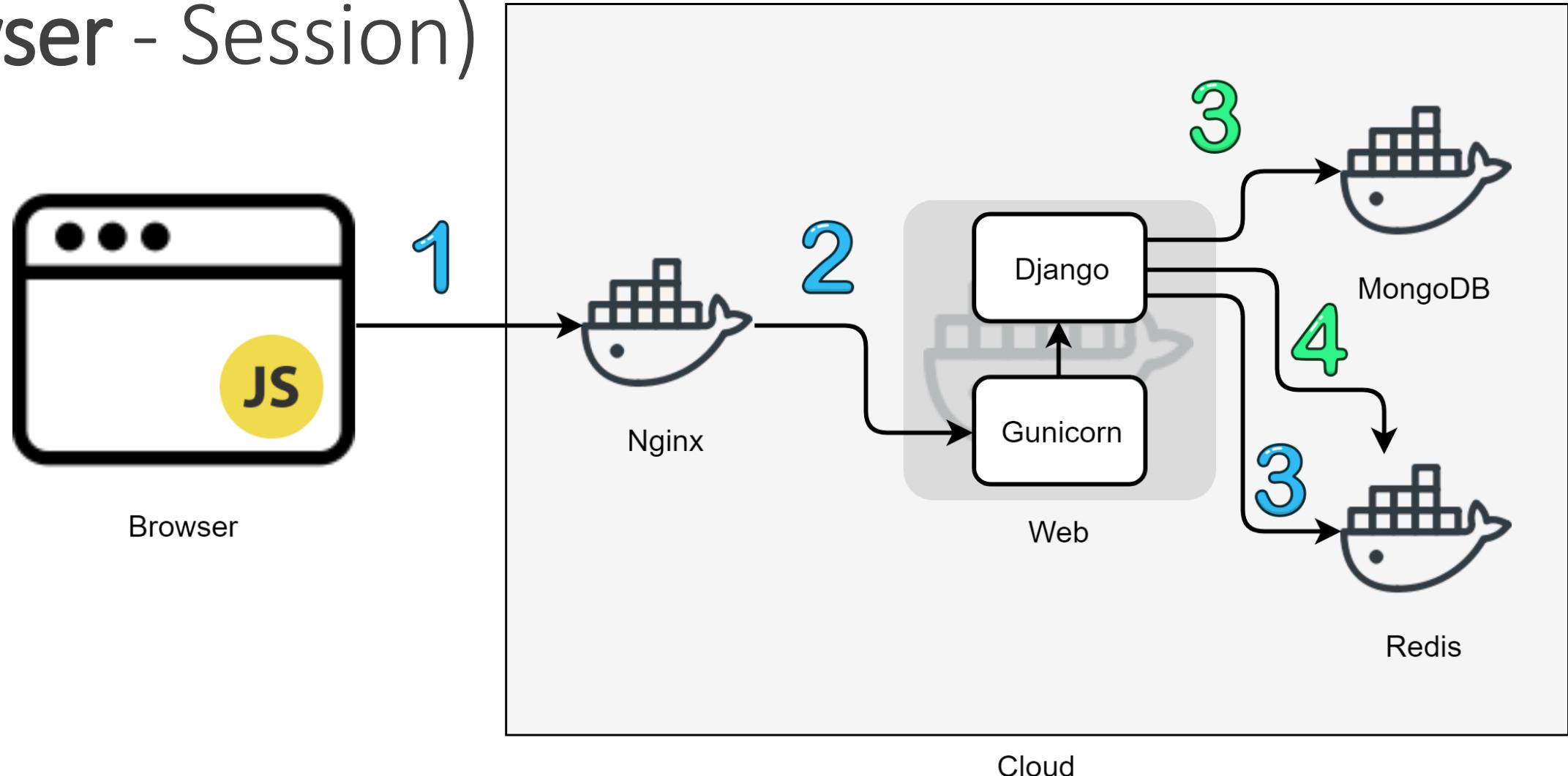
Secure Transmission (Edge - RTMP)



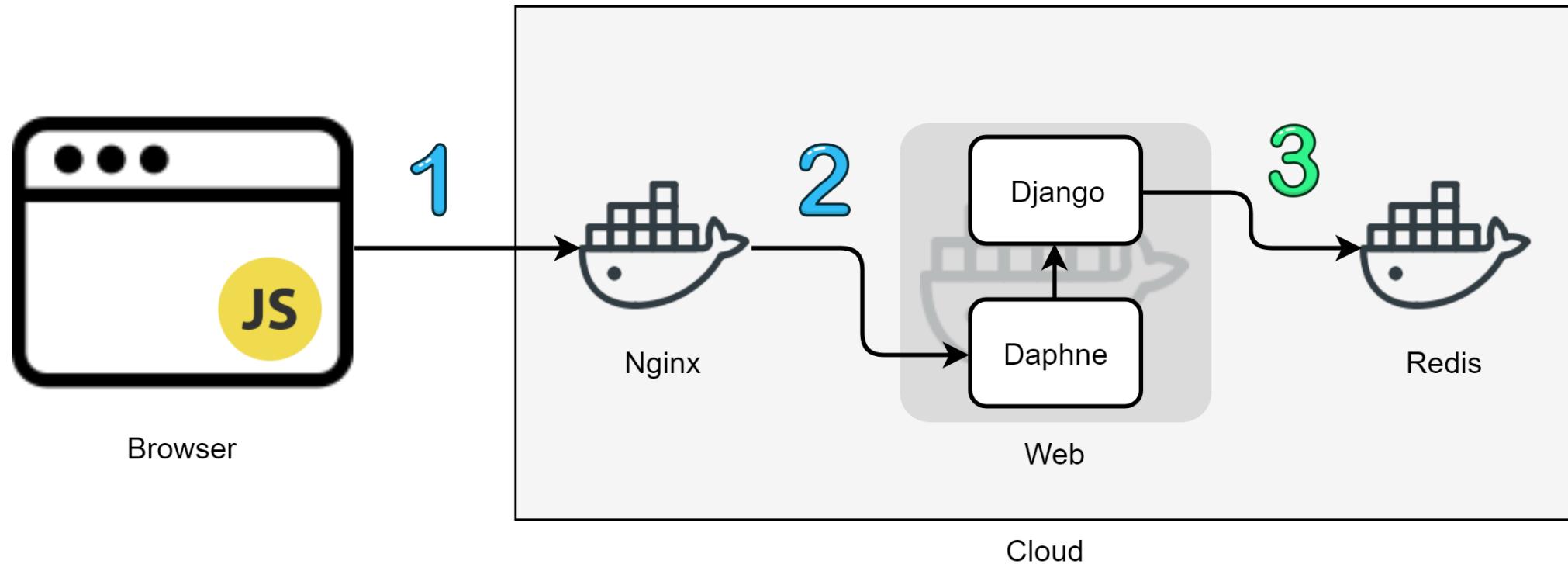
Secure Transmission (Edge - WebSocket)



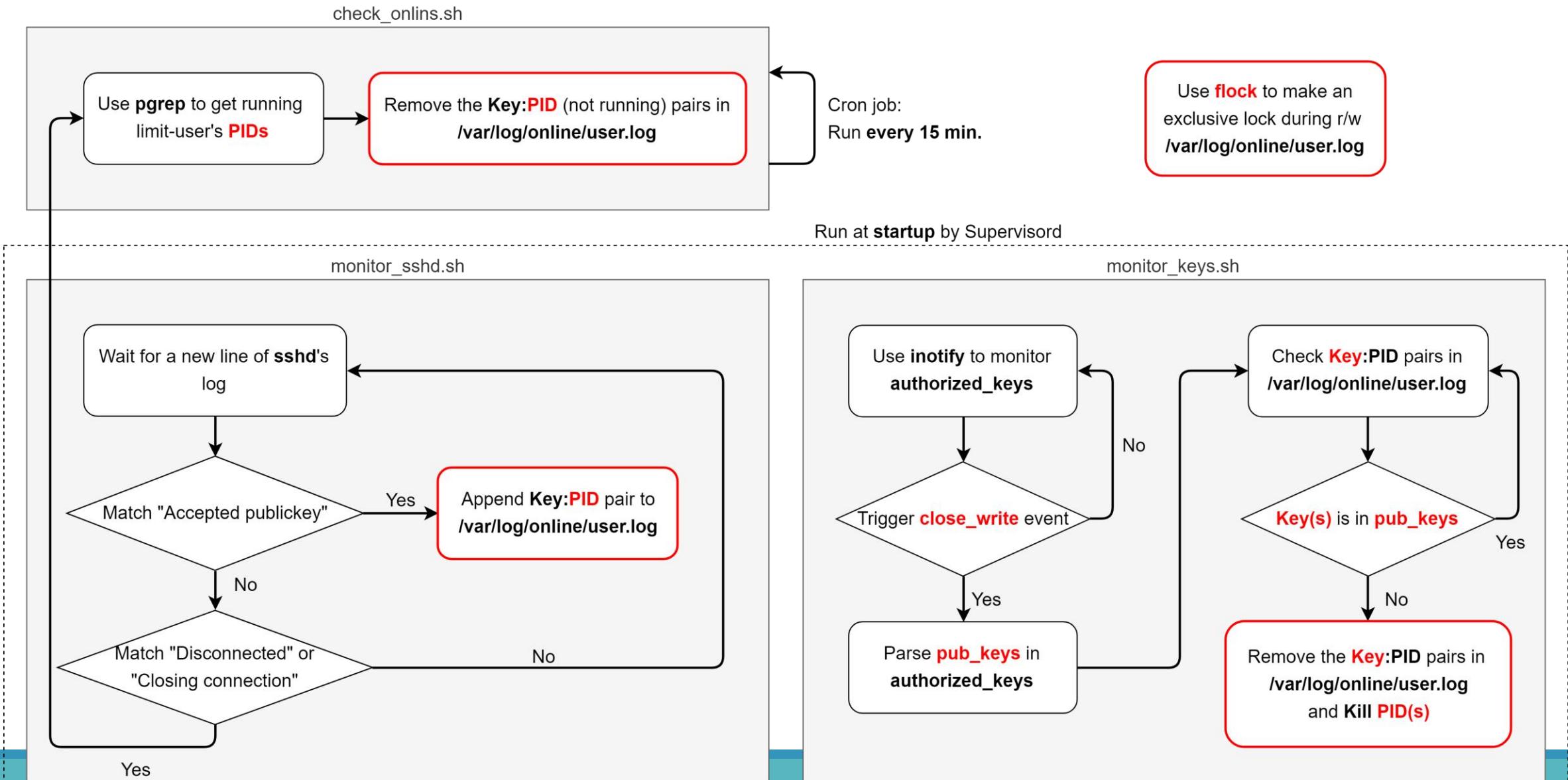
Secure Transmission (Browser - Session)



Secure Transmission (Browser - WebSocket)



Valid SSH User Checking



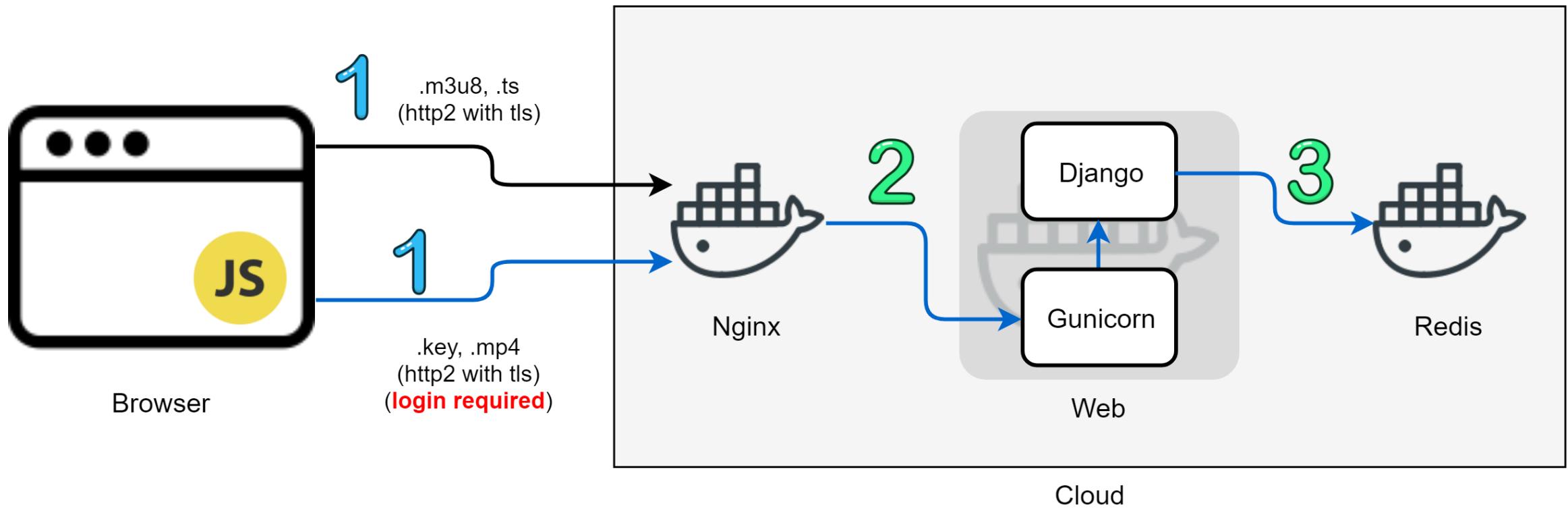
Certbot – HTTPS certificate Renewal

- Let's Encrypt
 - a free, automated, and open Certificate Authority
- Certbot
 - an extensible client for the Let's Encrypt CA
- Every **15** days → check if the certificate is expired in **30** days

Video Streaming (Nginx-RTMP)

- Listen on the internal port 1935 to receive **RTMP** stream with token check
- Generate HLS files
 - HLS playlist (.m3u8)
 - Encrypted media segment (.ts)
 - AES encryption key (.key)
- Record RTMP stream (FLV) and convert it into MP4 by FFmpeg
 - Maximum length of the historical video: 15 minutes
- Listen on the internal port 80 to receive control messages (sent by Django) to drop the RTMP connection (stream name=device id) if
 - Reset device **token**
 - **Delete** device

Secure Transmission (Browser - Video)



Video Streaming

HLS Encryption

The screenshot illustrates the process of HLS encryption. On the left, a video player window displays the HLS manifest file 'TX2_1.m3u8' with the following content:

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-MEDIA-SEQUENCE:206165
#EXT-X-TARGETDURATION:1
#EXT-X-KEY:METHOD=AES-128,URI="https://ssl.itlab.ee.ncku.edu.tw/key/TX2_1-206150.key",
IV=0x00000000000000000000000000000032546
#EXTINF:1.000,
TX2_1-206165.ts
#EXTINF:1.001,
TX2_1-206166.ts
#EXTINF:1.000,
TX2_1-206167.ts
```

Below the manifest, a message box shows a '403 Forbidden' error for the key URL: https://ssl.itlab.ee.ncku.edu.tw/key/TX2_1-206150.key.

On the right, a browser window titled 'Account | ITLab' shows a 403 Forbidden page with the text '403 Forbidden' and the word 'nginx' below it.

Below the browser is a 'Log In' form with fields for 'Username' and 'Password', a 'Remember Me' checkbox, and 'Log In' and 'Sign Up' buttons. There is also a 'Forgot password?' link.

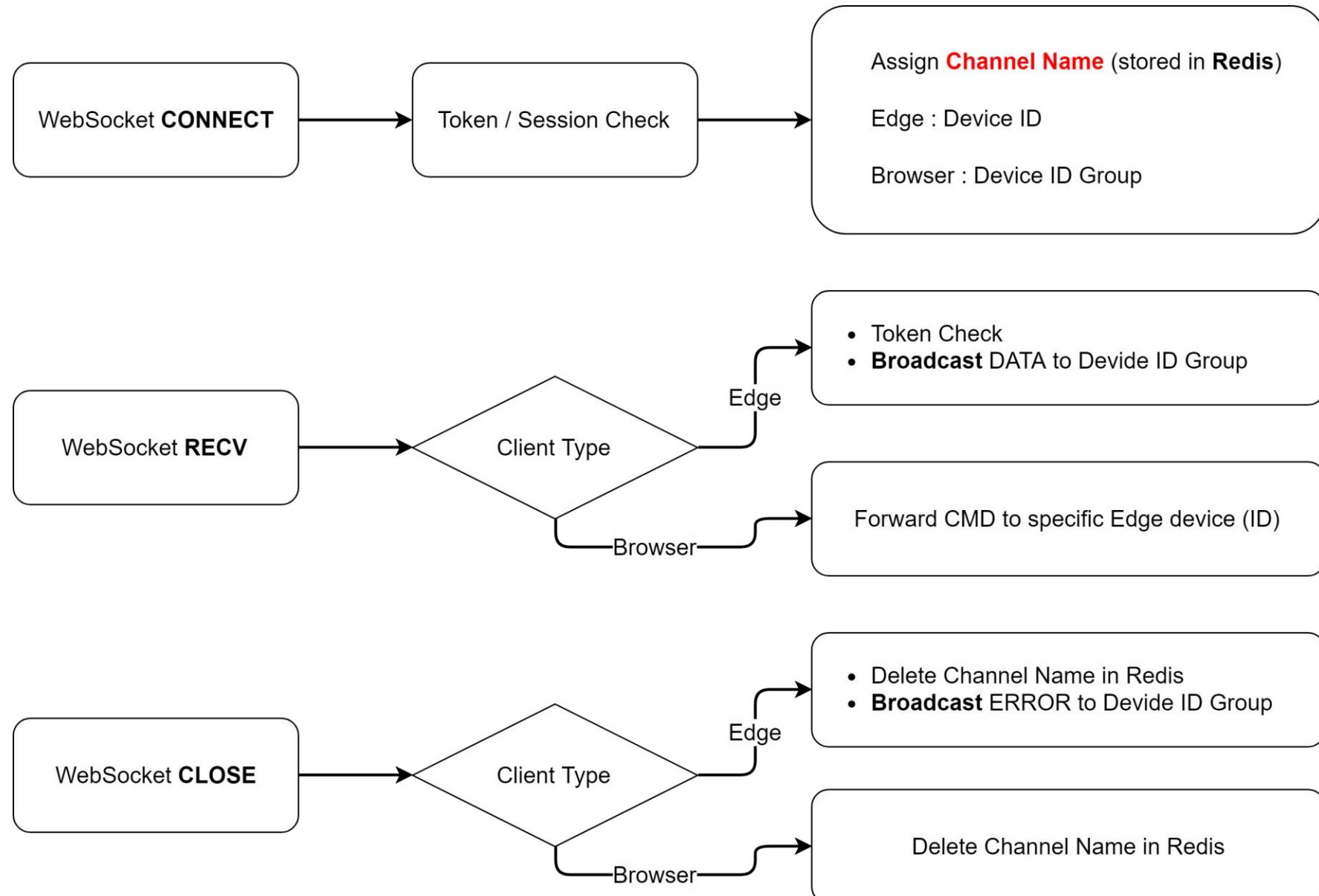
At the bottom right, there is a copyright notice: '© 2019 happyincent'.

Edge Service Handling (Cloud Module)

- Listen on the internal port 8001 to receive **WebSocket** connection
 - With token / session check
- Receive information sent **from** Edge and broadcast **to** Browsers (real-time)
- Receive commands sent **from** Browser and forward **to** the particular Edge (real-time)
 - Turn On/Off LED
 - Turn On/Off PIR
 - Change PIR timeout
- Send commands (LED/PIR) **to** the particular Edge every 15 min.
 - According to two timetables stored in MongoDB

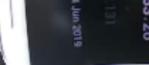
Edge Service Handling (Cloud Module)

WebSocket



Edge Service Handling (Cloud Module)

staff-ddl	(happyincent@gmail.com)																		
Log Out																			
Home / Device List / Device Info																			
<h1>TX2_1</h1> <hr/>																			
2019-06-24 23:32:40 +0800																			
<table border="1"> <tr> <td>Temperature</td> <td>26.10 °C</td> </tr> <tr> <td>Humidity</td> <td>51.50 %</td> </tr> <tr> <td>PM2.5</td> <td>4 µg/m³</td> </tr> <tr> <td>Loudness</td> <td>53.22 dB</td> </tr> <tr> <td>Light Intensity</td> <td>247.50 lux</td> </tr> <tr> <td>UV Intensity</td> <td>0.59 mW/cm²</td> </tr> <tr> <td>LED Status</td> <td> <input type="button" value="Off"/> <input checked="" type="button" value="PIR On"/> </td> </tr> <tr> <td>LED On / Off</td> <td>PIR On / Off</td> </tr> <tr> <td colspan="2"> LED Schedule Table PIR Schedule Table </td> </tr> </table>		Temperature	26.10 °C	Humidity	51.50 %	PM2.5	4 µg/m ³	Loudness	53.22 dB	Light Intensity	247.50 lux	UV Intensity	0.59 mW/cm ²	LED Status	<input type="button" value="Off"/> <input checked="" type="button" value="PIR On"/>	LED On / Off	PIR On / Off	LED Schedule Table PIR Schedule Table	
Temperature	26.10 °C																		
Humidity	51.50 %																		
PM2.5	4 µg/m ³																		
Loudness	53.22 dB																		
Light Intensity	247.50 lux																		
UV Intensity	0.59 mW/cm ²																		
LED Status	<input type="button" value="Off"/> <input checked="" type="button" value="PIR On"/>																		
LED On / Off	PIR On / Off																		
LED Schedule Table PIR Schedule Table																			

ee-dll	(happyincent@ee.ncku.edu.tw)
Log Out	
Home / Device List / Device Info	
TX2_1	
2019-06-24 23:33:30 +0800	
Temperature	26.20 °C
Humidity	51.80 %
PM2.5	4 µg/m ³
Loudness	53.71 dB
Light Intensity	248.33 lux
UV Intensity	0.50 mW/cm ²
LED Status	<input type="button" value="Off"/> <input checked="" type="button" value="PIR On"/>
	

ITLab Smart Street Light

≡

Edit Device TX2_1

Longitude

120.22283

Latitude

22.99672

SSH Public Key

ssh-rsa AAAAB3NzaC1yc2EAAAQAAQAI

Token

ef8a6d2d-cd2c-449f-a455-5d64

PIR Timeout (ms)

Waiting Edge ...

Update

Save

Back

ITLab Smart Street Light

Edit Device TX2_1

Longitude

Latitude

SSH Public Key

Token

Reset

PIR Timeout (ms)

Update

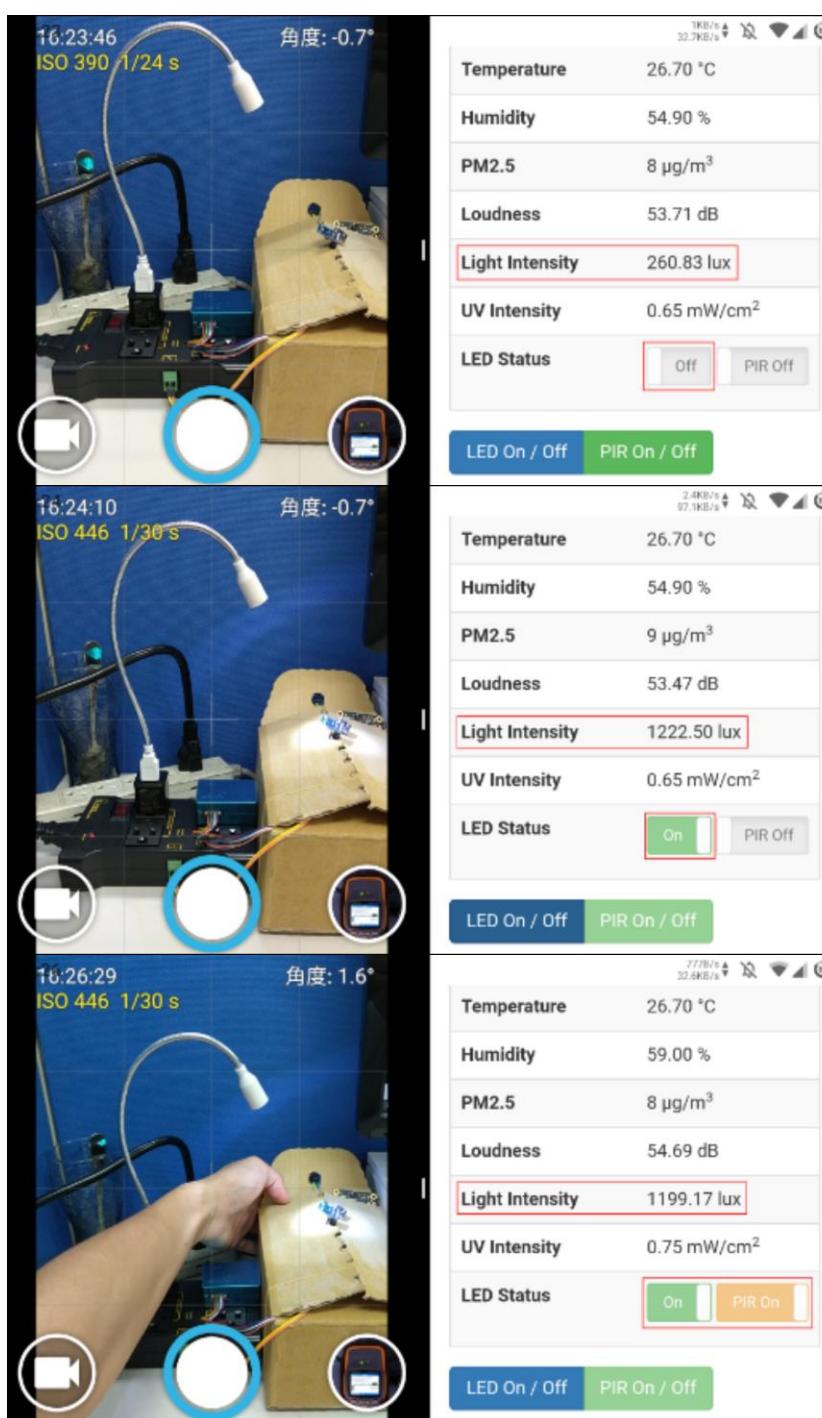
Save

Back

© 2019 happyincent

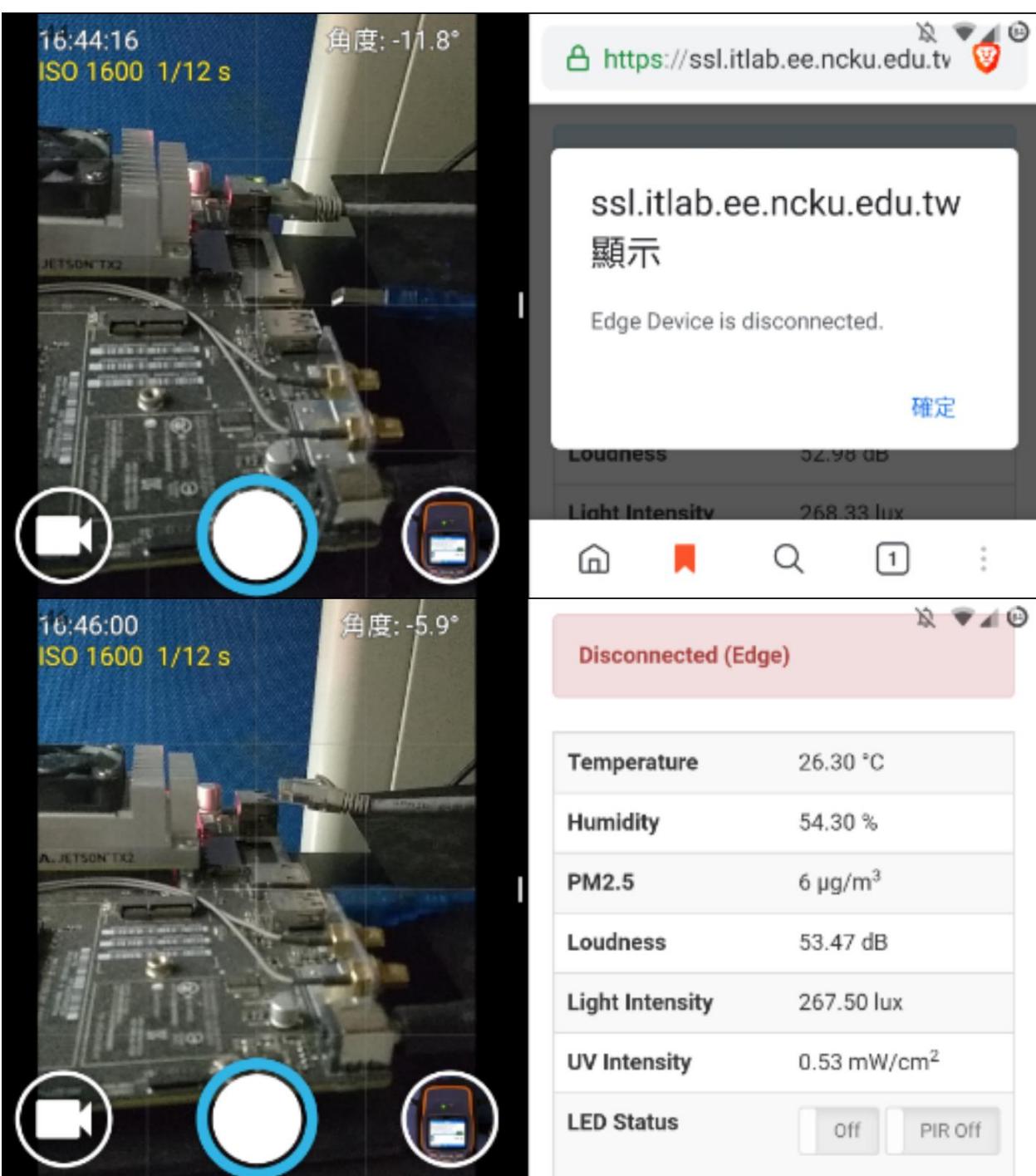
Edge Service Handling (Cloud Module)

- LED OFF & PIR OFF
- LED ON & PIR OFF
- LED ON & PIR ON

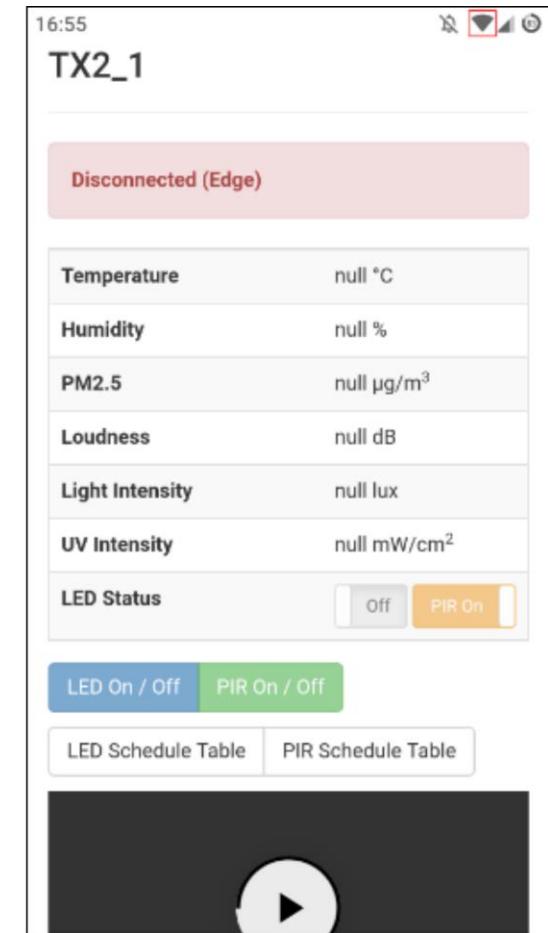
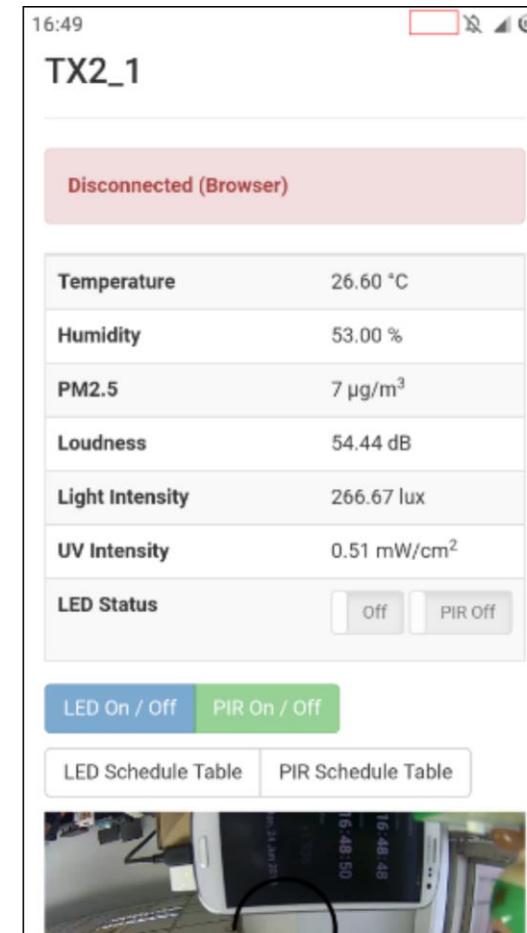
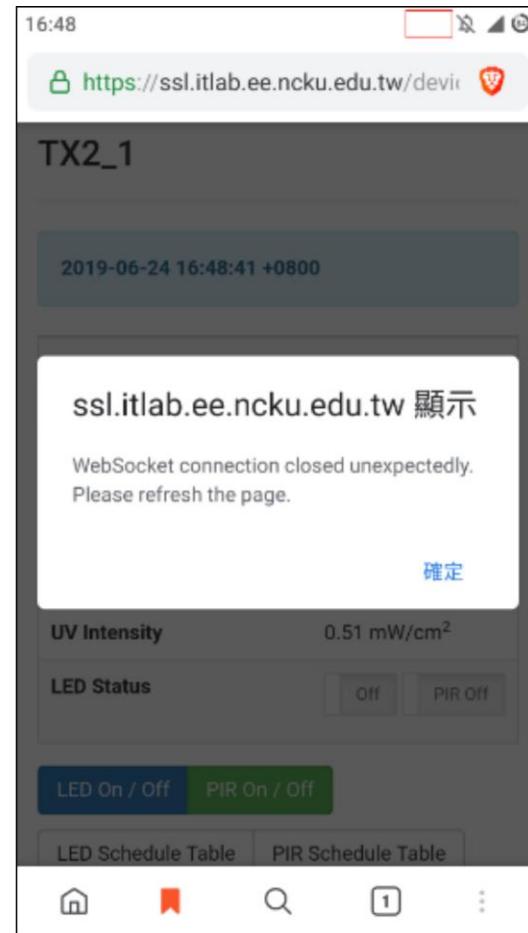
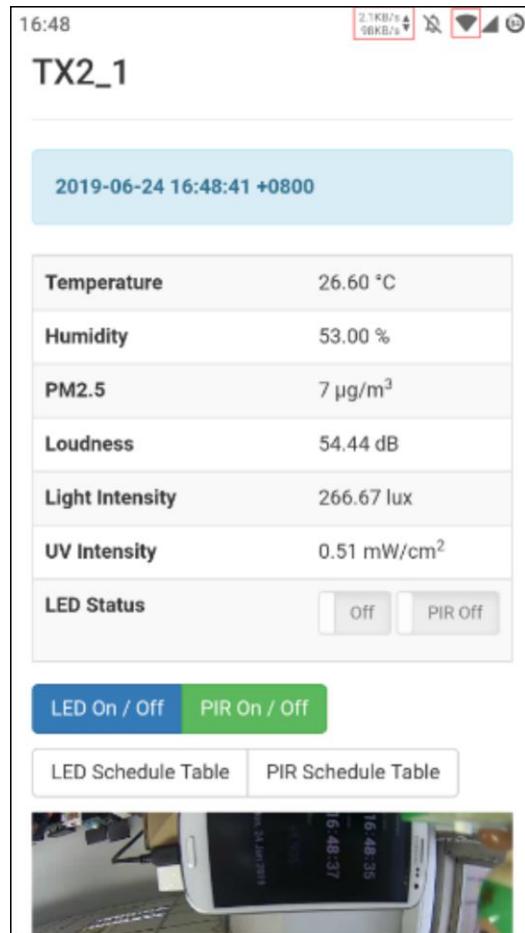


Edge Service Handling (Cloud Module)

- Remove USB-serial
- Remove RJ45



Edge Service Handling (Cloud Module)



Edge Service Handling (Cloud Module)

LED (PIR) ON/OFF Timetables (Send CMDs to Edge)

- Every 15 min.
- LED > PIR

The image displays two side-by-side screenshots of a mobile application interface for managing device schedules. Both screens have a header "ITLab Smart Street Light" and a navigation bar with "Home", "Device List", "Device Info", and "Schedule".

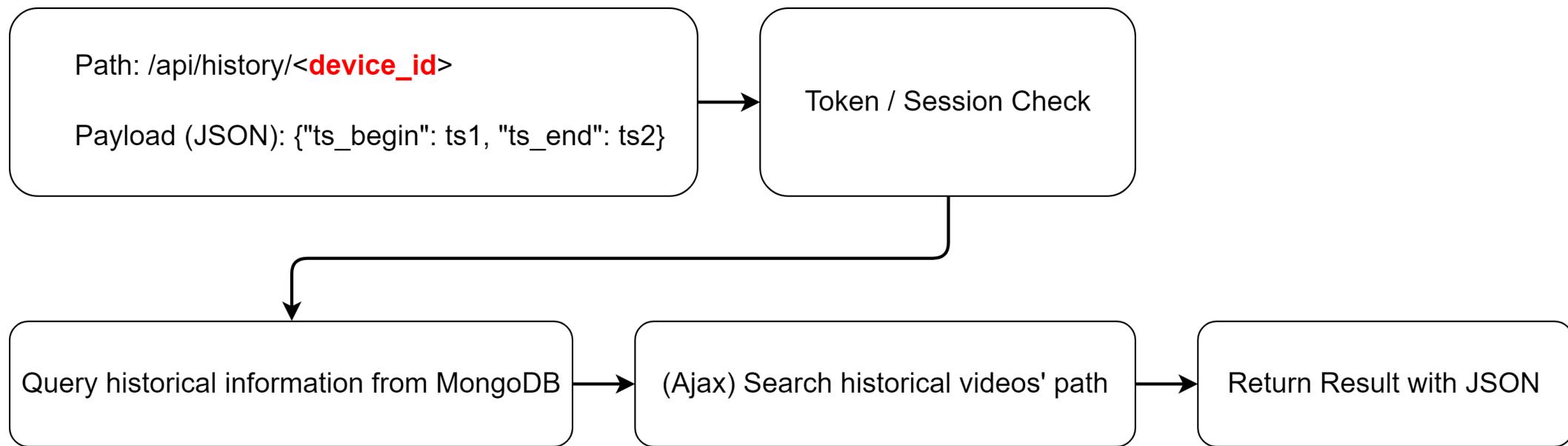
Left Screen (TX2_1 - LED ON Schedule): A message box states "Server would set up LED every 15 mins." Below are four buttons: "Restore" (blue), "Clear" (orange), "Save" (green), and "Back". The main area shows a 4x7 grid representing days (Monday to Thursday) and time slots (00:00 to 07:00). Blue shaded boxes indicate active LED schedules: Monday, Tuesday, and Wednesday show a 4:00-6:00 slot, while Thursday shows a 4:00-6:00 slot followed by a 04:00-06:00 slot.

Right Screen (TX2_1 - PIR ON Schedule): A message box states "Server would set up PIR every 15 mins. (note: Doing LED control force disable PIR)" Below are four buttons: "Restore" (blue), "Clear" (orange), "Save" (green), and "Back". The main area shows a 4x7 grid representing days (Monday to Thursday) and time slots (00:00 to 07:00). Blue shaded boxes indicate active PIR schedules: Monday, Tuesday, and Wednesday show a 0:00-4:00 slot, while Thursday shows a 0:00-4:00 slot followed by a 04:00-06:00 slot.

Historical Data Query – Cron Jobs

- Every 15 min.
 - Export current information from **Redis** into **MongoDB**
(loop all devices and check timestamp to avoid duplication)
- Every day
 - Time sync through *ntpd*
 - Delete the oldest 20% historical videos if the disk usage is over 80%
 - Delete historical documents in MongoDB which are older than 1 year

Historical Data Query API



Historical Data Query API (Browser - Ajax)

The screenshots show a series of four identical web pages for the device TX2_1, illustrating the historical data query API. Each page has a header "ITLab Smart Street Light" and a navigation bar "Home / Device List / Device History".

The first three screenshots show a date-time selector and a table of data. The fourth screenshot shows a single row of data.

Screenshots 1, 2, and 3: A modal dialog box displays the instruction "Get records for one hour from the selected DateTime." Below it is a date-time picker. The calendar shows June 2019, with the 18th highlighted. The time picker shows 3:00. The table below contains four rows of data, each with columns for Light Intensity, UV Intensity, LED Status, and Load (partially visible). The first three rows have red boxes around their Light Intensity and UV Intensity columns. The fourth row has a red box around its Light Intensity column.

Screenshot 4: The same interface, but the table only contains a single row of data. This row also has a red box around its Light Intensity column.

Light Intensity	UV Intensity	LED Status	Load
62.5 lux	0.67 mW/cm ²	off	
62.5 lux	0.55 mW/cm ²	off	
62.5 lux	0.5 mW/cm ²	off	
62.5 lux	0.67 mW/cm ²	off	
1133.33 lux	0.54 mW/cm ²	on	
1133.33 lux	0.49 mW/cm ²	on	
1133.33 lux	0.5 mW/cm ²	on	
1134.17 lux	0.57 mW/cm ²	on	

Historical Data Query API (Browser - Ajax)

Device History | ITLab SSL

https://ssl.itlab.ee.ncku.edu.tw/history/TX2_1

TX2_1

Get records for **one hour** from the selected **DateTime**.

2019-06-30 18:00:00

Time	Temperature	Humidity	PM2.5	Loudness	Light Intensity	UV Intensity	LED Status	Load Video
2019-06-30 18:00:29 +0800	25.6 °C	53.9 %	3 µg/m ³	54.2 dB	241.67 lux	0.62 mW/cm ²	off	<button>Load</button>
2019-06-30 18:15:30 +0800	25.7 °C	54.1 %	2 µg/m ³	53.47 dB	239.17 lux	0.61 mW/cm ²	off	<button>Load</button>
2019-06-30 18:30:30 +0800	25.6 °C	52.7 %	3 µg/m ³	55.66 dB	1500.83 lux	0.82 mW/cm ²	on	<button>Load</button>
2019-06-30 18:45:31 +0800	25.4 °C	50 %	2 µg/m ³	54.44 dB	1476.67 lux	0.67 mW/cm ²	on	<button>Load</button>

The image shows a smartphone held in a hand. The screen displays a video feed of an indoor environment. Overlaid on the video are several data elements: a timestamp '18:12:49' at the top right, a date 'Sun, 30 Jun 2019' below it, a temperature reading '+0.659' on the left, and a light intensity reading '1500.83 lux' on the right. At the bottom of the phone's screen, there is a video player interface showing a play button, the time '0:03 / 14:59', and other control icons.

© 2019 happyincent

Historical Data Query API (Open Data)

/api/device

{

“user”: username,

“token”: token

}

ITLab Smart Street Light Device List Open Data API About US ee-ddl (happyincent@ee.ncku.edu.tw) Log Out

Home / Open Data API

API token (TTL: 00:09:59)
2f951e4e-0f1f-4354-bd47-9b0e3a37a69d Reset

API usage

POST	/api/device	Query device list
Description	List all devices	Request
Method	POST	
URI	https://ssl.itlab.ee.ncku.edu.tw/api/device	
Request header	• Content-Type : application/json; charset=utf-8	
Request body	• "user" : Username (String) • "token" : API token (String)	
Request body example	{ "user": "itlab", "token": "54c51ef2-cb2f-4000-9500-6d4af31e7a02" }	
Sample code	api_device.py.txt	Success Response
Response CODE	200 (OK) : Query Successfully	
Data type	JSON	
Response message parameters	• "id" : Device ID (String) • "longitude" : Longitude (Double) • "latitude" : Latitude (Double)	
Response example	[{"id": "TX2_1", "longitude": 120.22283, "latitude": 22.99672}, {"id": "RPI_1", "longitude": 120.22283, "latitude": 22.99872}, ...]	Error Response
Response CODE	403 (Forbidden) : Token Authentication Fail	

POST /api/history/\${Device_ID} Query historical information

© 2019 happyincent

Historical Data Query API (Open Data)

`/api/history/${Device_ID}`

{

`“user”: username,`

`“token”: token,`

`“ts_begin”: 1560801000,`

`“ts_end”: 1560803000,`

}

ITLab Smart Street Light Device List Open Data API About US ee-ddl (happyincent@ee.ncku.edu.tw) Log Out

Home / Open Data API

API token (TTL: 00:09:11)
2f951e4e-0f1f-4354-bd47-9b0e3a37a69d Reset

API usage

POST	/api/device	Query device list
------	-------------	-------------------

POST	/api/history/\${Device_ID}	Query historical information
Description		
Get historical information within a period of time		
Request		
Method	POST	
URI	https://ssl.itlab.ee.ncku.edu.tw/api/history/\${Device_ID}	
Path parameters	• Device_ID : Device ID of the streetlight	
Request header	• Content-Type : application/json; charset=utf-8	
Request body	<ul style="list-style-type: none">• “user” : Username (String)• “token” : API token (String)• “ts_begin” : Unix timestamp - Start time (Integer)• “ts_end” : Unix timestamp - End time (Integer)	
Request body example	{“user”:“itlab”, “token”:“54c51ef2-cb2f-4000-9500-6d4af31e7a02”, “ts_begin”:“1560801000”, “ts_end”:“1560803000”}	
Sample code	api_history.py.txt	
Success Response		
Response CODE	200 (OK) : Query Successfully	
Data type	JSON	
Response message parameters	<ul style="list-style-type: none">• “id” : id (Integer)• “device_id” : Device ID (String)• “temperature” : °C (Double)• “humidity” : % (Double)• “pm25” : µg/m³ (Double)• “loudness” : dB (Double)• “light_intensity” : lux (Double)• “uv_intensity” : mW/cm² (Double)• “led_status” : true or false (Boolean)• “timestamp” : date and time (ISO 8601)	
Response example	[{"id": 61746, "device_id": "TX2_1", "temperature": 24.5, "humidity": 46.7, "pm25": 6.0, "loudness": 54.69, "light_intensity": 62.5, "uv_intensity": 0.67, "led_status": false, "timestamp": "2019-06-18T03:59:59+08:00"}, {"id": 61766, "device_id": "TX2_1", "temperature": 24.5, "humidity": 46.6, "pm25": 6.0, "loudness": 53.71, "light_intensity": 1133.33, "uv_intensity": 0.54, "led_status": true, "timestamp": "2019-06-18T04:15:00+08:00"}]	
Error Response		
Response CODE	403 (Forbidden) : Token Authentication Fail	
	404 (Not Found) : Device_ID has no historical information in the period of time	

Outline

- Introduction
- Background and Related works
- System Design
 - Cloud Management System
 - Edge Services Orchestrator
- Experiment and Verification
- Conclusion and Future works

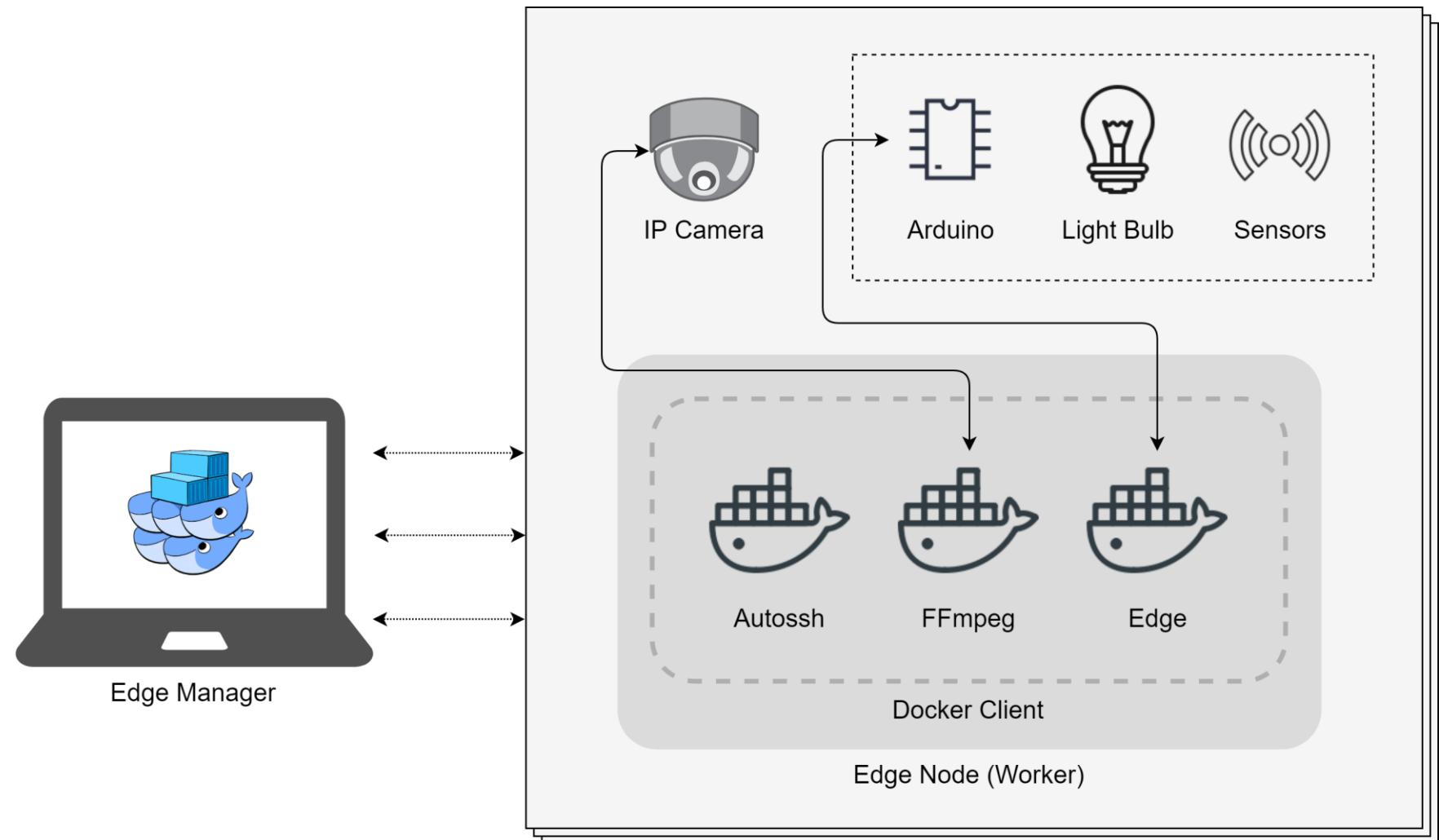
Edge Services Orchestrator

- Edge Nodes Architecture
 - Overview
 - Docker images, containers
- Edge Services
 - Secure Transmission Module
 - Video Streaming Module
 - Edge Service Module
- Edge Nodes Deployment
 - Single Node Deployment
 - Cluster Nodes Deployment

Edge Services Orchestrator

- Edge Nodes Architecture
 - Overview
 - Docker images, containers
- Edge Services
 - Secure Transmission Module
 - Video Streaming Module
 - Edge Service Module
- Edge Nodes Deployment
 - Single Node Deployment
 - Cluster Nodes Deployment

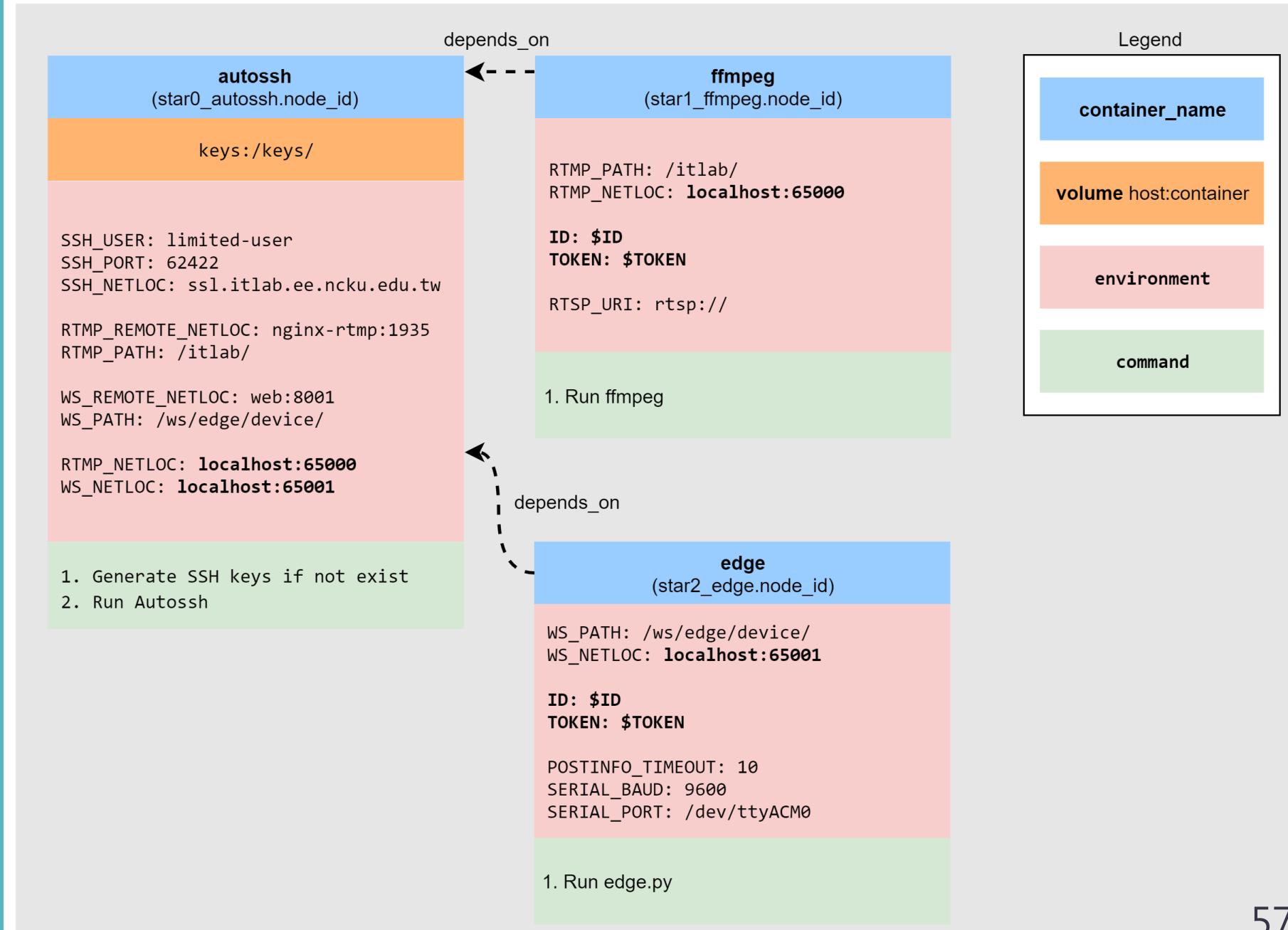
Edge Nodes - Overview



Edge Nodes - Docker Images

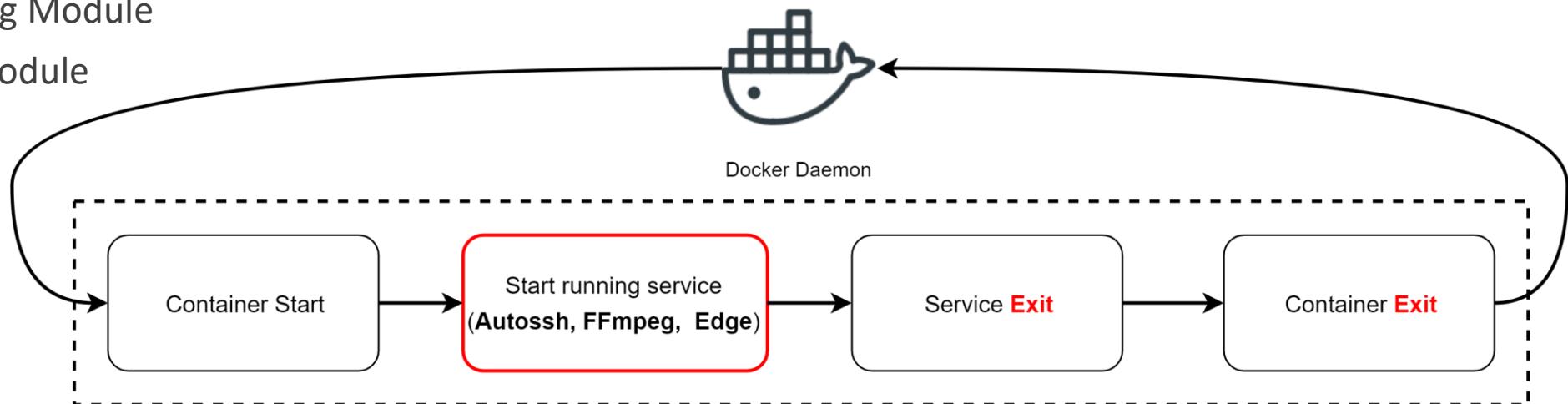
Info	Image	Base Image	Size (amd64)	Size (arm64)	Size (armv7)
Autossh		alpine:3.9	11.5MB	11.2MB	8.71MB
FFmpeg		alpine:3.9	57MB	47.9MB	35.6MB
Edge		python:3.7.3-alpine3.9	93.7MB	100MB	78MB

Edge Nodes - Docker Containers

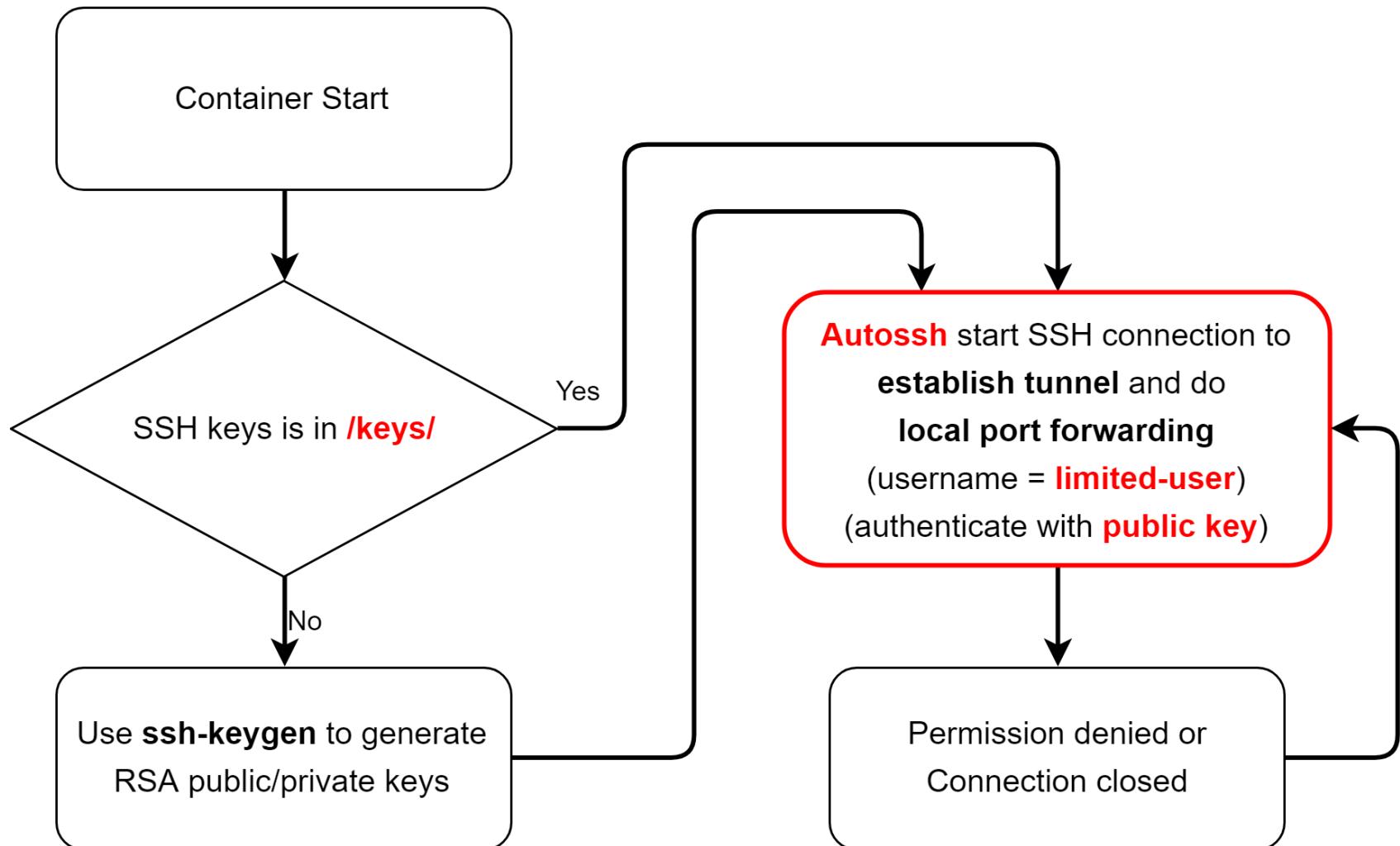


Edge Services Orchestrator

- Edge Nodes Architecture
 - Overview
 - Docker images, containers
- Edge Services
 - Secure Transmission Module
 - Video Streaming Module
 - Edge Service Module
- Edge Nodes Deployment
 - Single Node Deployment
 - Cluster Nodes Deployment



Secure Transmission Module



Video Streaming Module

- RTSP → FFmpeg (with token) → (SSH Tunnel) → Nginx-RTMP

```
ffmpeg \
-y -loglevel warning \
-use_wallclock_as_timestamps 1 \
-flags global_header \
-fflags +genpts \
-r 15 -rtsp_transport tcp \
-i $$RTSP_URI \
-vcodec copy -an \
-bsf:v h264_mp4toannexb \
-tune zerolatency \
-f flv -rtmp_live live -rtmp_buffer 0 -timeout 0 \
rtmp://{$RTMP_NETLOC}{$RTMP_PATH}$$ID?token=$$TOKEN
```

Edge Service Module – Edge Container

- USB-serial → Edge → WebSocket (with token) → (SSH Tunnel) → Web
(Threads)
- **SerialHandler** (Arduino to Edge)
 - Forward **INFO** to Cloud
- **PostInfoHandler** (Edge to Arduino)
 - Ask Arduino for the latest **INFO** every 10 s
- **MessageHandler** (Cloud to Edge)
 - Forward **CMDs** to Arduino

Edge Service Module – Arduino

(LOOP)

- Read Serial **CMDs** (Edge to Arduino)
 - Turn On/Off LED
 - Turn On/Off PIR
 - Change PIR timeout
- PIR Control (if enable)
 - Sensed → Turn On LED → Turn Off LED (after timeout)
- Query Sensors' value (every 1 s)

Edge Services Orchestrator

- Edge Nodes Architecture
 - Overview
 - Docker images, containers
- Edge Services
 - Secure Transmission Module
 - Video Streaming Module
 - Edge Service Module
- **Edge Nodes Deployment**
 - Single Node Deployment
 - Cluster Nodes Deployment

Single Node Deployment

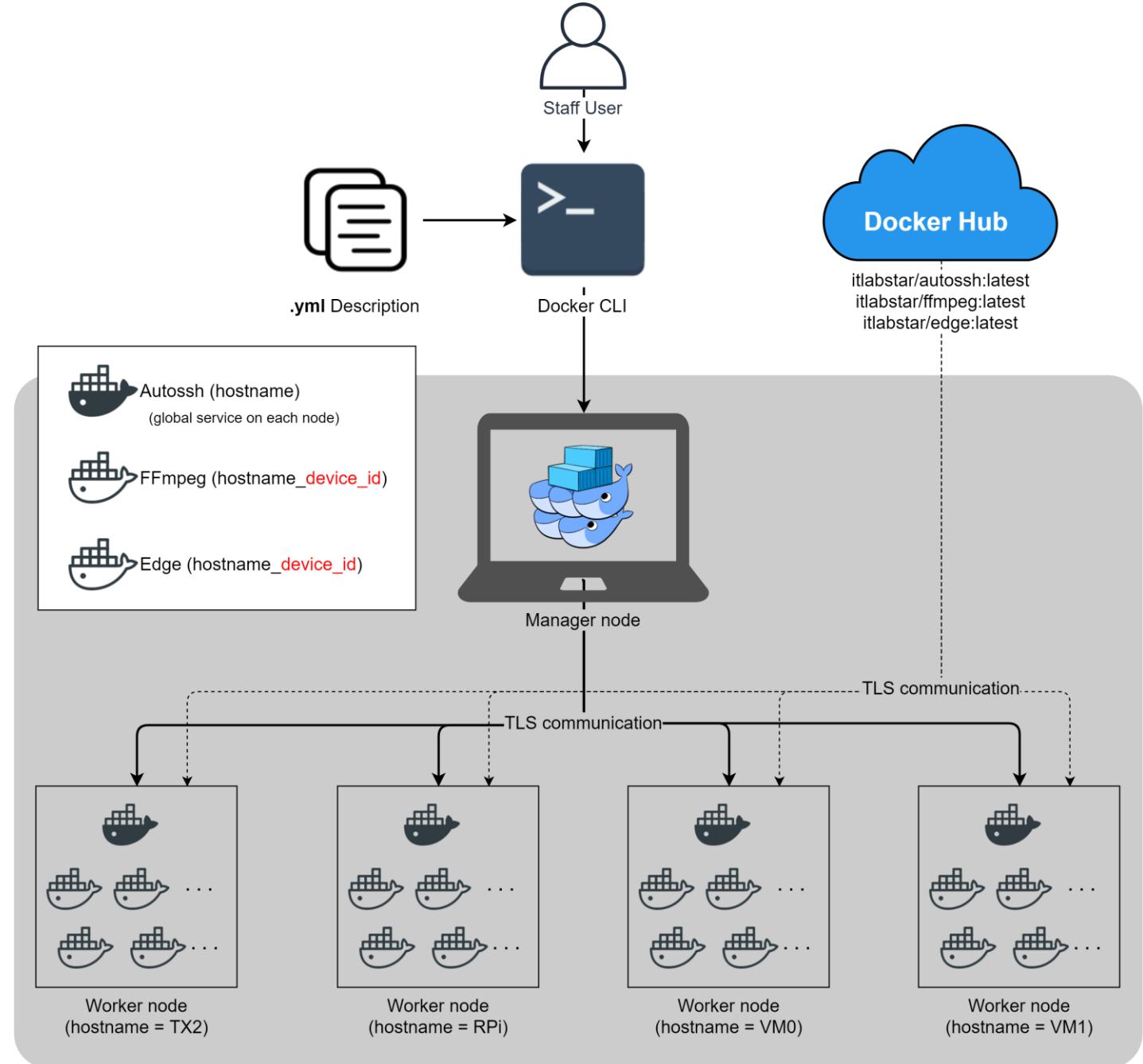
- Build images locally
 - Dockerfiles
- Run containers
 - Compose files (.ymls), Environment files (.env, star-single.env)

```
nvidia@TX2:~/Desktop/star-edge$ echo "$(cat .env)"  
#### Edge Service Parameters ####  
SSH_USER=limited-user  
SSH_PORT=62422  
SSH_NETLOC=ssl.itlab.ee.ncku.edu.tw  
  
RTMP_REMOTE_NETLOC=nginx-rtmp:1935  
RTMP_PATH=/itlab/  
  
WS_REMOTE_NETLOC=web:8001  
WS_PATH=/ws/edge/device/  
  
RTMP_NETLOC=localhost:65000  
WS_NETLOC=localhost:65001
```

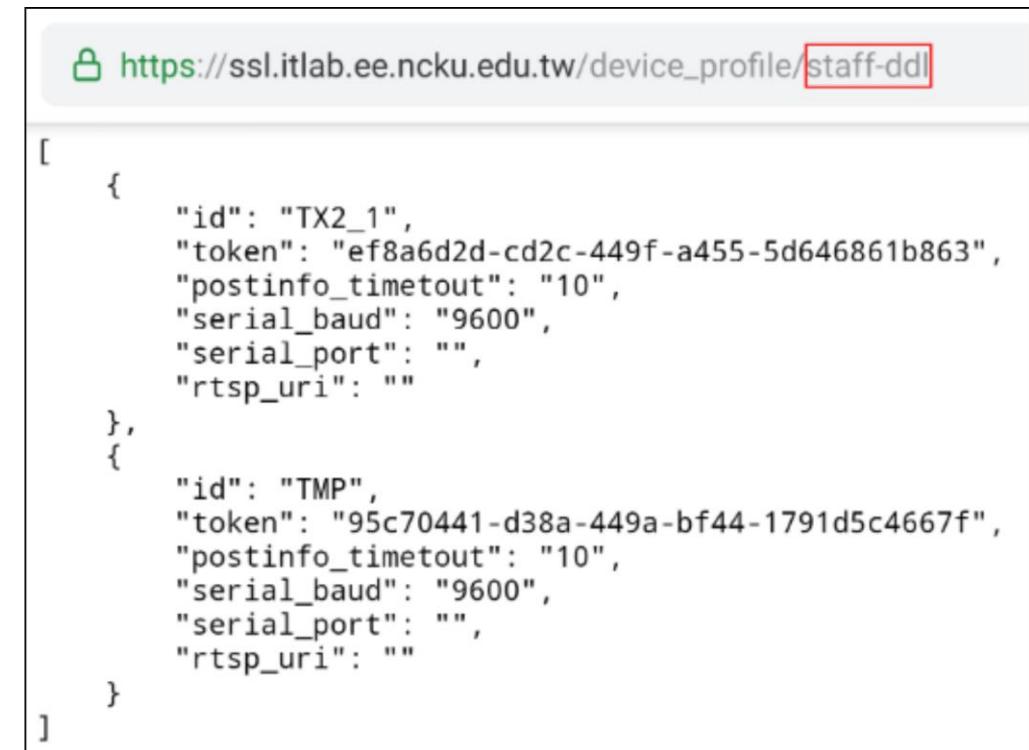
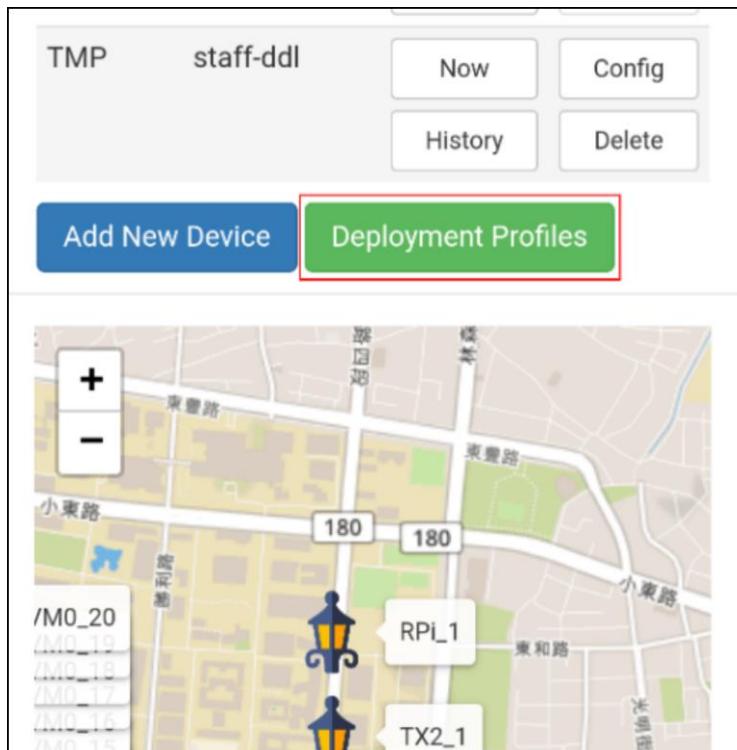
```
nvidia@TX2:~/Desktop/star-edge$ echo "$(cat star-single.env)"  
#### Device Parameters ####  
ID=TX2_1  
TOKEN=ef8a6d2d-cd2c-449f-a455-5d646861b863  
  
POSTINFO_TIMEOUT=10  
SERIAL_BAUD=9600  
SERIAL_PORT=/dev/ttyACM0  
  
RTSP_URI=rtsp://[REDACTED]@10.27.164.152/live3.sdp
```

Cluster Nodes Deployment

- Docker Swarm
- Docker Compose (.ymls)
- Docker buildx
(cross-platform images)
- Docker Hub



Compose File Generator



Compose File Generator

```
itlab@X302LJ:~/Desktop/star-edge$ rm node/yml/*
itlab@X302LJ:~/Desktop/star-edge$ ls node/profile/
RPi.json  TX2.json  VM0.json  VM1.json
itlab@X302LJ:~/Desktop/star-edge$ echo "$(cat node/profile/TX2.json)"
{
    "NODE_NAME": "TX2",
    "DEVICES": [
        {
            "id": "TX2_1",
            "token": "ef8a6d2d-cd2c-449f-a455-5d646861b863",
            "postinfo_timetout": "10",
            "serial_baud": "9600",
            "serial_port": "/dev/ttyACM0",
            "rtsp_uri": "rtsp://[REDACTED]@10.27.164.152/live3.sdp"
        },
        {
            "id": "TMP",
            "token": "95c70441-d38a-449a-bf44-1791d5c4667f",
            "postinfo_timetout": "10",
            "serial_baud": "9600",
            "serial_port": "",
            "rtsp_uri": "rtsp://[REDACTED]/proxyStream"
        }
    ]
}
itlab@X302LJ:~/Desktop/star-edge$ ./node/generate.sh `ls $PWD/node/profile/*.json`  
Gernerating YAMLs from the profile: /home/itlab/Desktop/star-edge/node/profile/RPi.json ...
Gernerating YAMLs from the profile: /home/itlab/Desktop/star-edge/node/profile/TX2.json ...
Gernerating YAMLs from the profile: /home/itlab/Desktop/star-edge/node/profile/VM0.json ...
Gernerating YAMLs from the profile: /home/itlab/Desktop/star-edge/node/profile/VM1.json ...
itlab@X302LJ:~/Desktop/star-edge$ tree -P "TX2*" node/yml/
node/yml/
└── TX2_TMP-edge.yml
    ├── TX2_TMP-ffmpeg.yml
    └── TX2_TX2_1-edge.yml
        └── TX2_TX2_1-ffmpeg.yml
```

Swarm Visualizer

The image shows a screenshot of the Swarm Visualizer application. At the top right is a white whale icon representing Docker. Below it is a search bar labeled "filter containers". The main area displays a grid of container status cards for different hosts:

- RPi worker**: 0.854G RAM, armv7l/linux. Contains cards for star1_RPi_RPi_1-ffmpeg and star0_autossh.
- TX2 worker**: 7.670G RAM, aarch64/linux. Contains cards for star0_autossh and star1_TX2_TX2_1-ffmpeg.
- VM0 worker**: 11.727G RAM, x86_64/linux. Contains cards for star0_autossh, star1_VM0_VM0_7-ffmpeg, star1_VM0_VM0_5-ffmpeg, and star2_VM0_VM0_1-edge.
- VM1 worker**: 0.962G RAM, x86_64/linux. Contains cards for star1_VM1_VM1_6-ffmpeg and star2_VM1_VM1_5-edge.
- X302LJ manager**: 7.695G RAM, x86_64/linux. Contains cards for manager_visualizer, star1_VM1_VM1_5-ffmpeg, manager_live555proxy, and star2_VM1_VM1_4-edge.

Each card provides details about the container's name, image, tag, command, update time, and state. A red border highlights the "armv7l/linux" and "aarch64/linux" labels, indicating specific host architectures.

Outline

- Introduction
- Background and Related works
- System Design
- **Experiment and Verification**
- Conclusion and Future works

Environment

(Cloud Server)	HP Z600	(Other Devices)	Description
CPU	16 x 2.27 GHz	Arduino Mega 2560	Microcontroller
RAM	24 GB	VIVOTEK FD8169A	IP Camera
OS	Xubuntu 16.04.6	ASUS Laptop X302LJ	<ul style="list-style-type: none"> • Swarm Manager • RTSP Proxy Server
Software	Docker 18.09.7, Docker Compose 1.24.0		

(Edge Nodes)	Nvidia Jetson TX2	Raspberry Pi 3B	VM0 (ESXi 6.0)	VM1 (ESXi 5.5)
CPU	2 x2 + 4x 2 GHz	4 x 1.2 GHz	8 x 3.7 GHz	8 x 3.4 GHz
RAM	8 GB	1 GB	12 GB	2 GB
OS	Ubuntu 16.04.6	Raspbian 9.9	Xubuntu 16.04.6	Ubuntu server 18.04.2
Software	Docker 18.09.7	Docker 18.09.0	Docker 18.09.6	Docker 18.09.6

Backup & Restore Cloud Server

- Start time ≈ 21 s
- Stop time ≈ 35 s
- Size (with DB & Code)
≈ 500 MB

```
itlab@Z600:~/Desktop/star-eval$ ./backup.sh
Stop all Docker containers ...

real    0m34.724s
user    0m1.165s
sys     0m0.119s

OK

Backup star project ...

real    0m15.012s
user    0m14.671s
sys     0m1.510s

OK

Backup Docker images ...

real    0m29.452s
user    0m53.618s
sys     0m2.806s

OK

List tar files ...
213M  star.tar.gz
74M   certbot:star.tar.gz
50M   mongo:star.tar.gz
38M   nginx-rtmp:star.tar.gz
7.3M   nginx:star.tar.gz
17M   redis:star.tar.gz
18M   sshd:star.tar.gz
83M   web:star.tar.gz
itlab@Z600:~/Desktop/star-eval$ [REDACTED]

itlab@Z600:~/Desktop/star-eval$ ./restore.sh
Remove star project ...
OK

Remove Docker images ...
OK

Restore star project ...

real    0m4.318s
user    0m4.302s
sys     0m1.842s

OK

Restore Docker images ...

real    0m11.014s
user    0m10.863s
sys     0m1.755s

OK

Start all Docker containers ...

real    0m20.819s
user    0m1.326s
sys     0m0.099s

OK
itlab@Z600:~/Desktop/star-eval$ [REDACTED]
```

Secure Transmission Verification

Dataflow	Transmission	Encryption	Authentication
Cloud → Browser (all requests) (hls key, vod mp4)	TCP / HTTP	HTTPS	Session
Cloud → Browser (edge service)	TCP / WebSocket	HTTPS	Session
Edge → Cloud (edge service)	TCP / WebSocket	SSH tunnel	Token
Edge → Cloud (streaming)	TCP / RTMP	SSH tunnel	Token
IP Camera → Edge (streaming)	TCP / RTSP	None (LAN)	Fixed password

Netfilter Logs

- WAN to Container (Internet → SSHD)

```
itlab@Z600:~$ cat log | sed -e 's|ctc/iptable/||g' -e 's|wtc/iptable/||g' -e 's/PHYSIN=[^ ]*//g' -e 's/PHYSOUT=[^ ]*//g' -e '^ ]*//g' -e 's/ID=[^ ]*//g' -e 's/DF[^ ]*//g' -e 's/ \+/,/g' | column -t -s,
```

mangle-PREROUTE	IN=enp1s0	OUT=	SRC=10.27.164.153	DST=140.116.164.152	PROTO=TCP	SPT=59652	DPT=62422
nat-PREROUTE	IN=enp1s0	OUT=	SRC=10.27.164.153	DST=140.116.164.152	PROTO=TCP	SPT=59652	DPT=62422
mangle-FORWARD	IN=enp1s0	OUT=br-73672de335db	SRC=10.27.164.153	DST=172.18.0.5	PROTO=TCP	SPT=59652	DPT=22
filter-forward	IN=enp1s0	OUT=br-73672de335db	SRC=10.27.164.153	DST=172.18.0.5	PROTO=TCP	SPT=59652	DPT=22
mangle-POSTROUTER	IN=	OUT=br-73672de335db	SRC=10.27.164.153	DST=172.18.0.5	PROTO=TCP	SPT=59652	DPT=22
nat-POSTROUTER	IN=	OUT=br-73672de335db	SRC=10.27.164.153	DST=172.18.0.5	PROTO=TCP	SPT=59652	DPT=22

- Container to Container (SSHD → Web, Web → Redis)

```
itlab@Z600:~$ cat log | sed -e 's|ctc/iptable/||g' -e 's|wtc/iptable/||g' -e 's/PHYSIN=[^ ]*//g' -e 's/PHYSOUT=[^ ]*//g' -e '^ ]*//g' -e 's/ID=[^ ]*//g' -e 's/DF[^ ]*//g' -e 's/ \+/,/g' | column -t -s,
```

mangle-PREROUTE	IN=br-73672de335db	OUT=	SRC=172.18.0.5	DST=172.18.0.7	PROTO=TCP	SPT=35806	DPT=8001
filter-forward	IN=br-73672de335db	OUT=br-73672de335db	SRC=172.18.0.5	DST=172.18.0.7	PROTO=TCP	SPT=35806	DPT=8001
mangle-POSTROUTER	IN=	OUT=br-73672de335db	SRC=172.18.0.5	DST=172.18.0.7	PROTO=TCP	SPT=35806	DPT=8001
mangle-PREROUTE	IN=br-73672de335db	OUT=	SRC=172.18.0.7	DST=172.18.0.6	PROTO=TCP	SPT=45866	DPT=6379
filter-forward	IN=br-73672de335db	OUT=br-73672de335db	SRC=172.18.0.7	DST=172.18.0.6	PROTO=TCP	SPT=45866	DPT=6379
mangle-POSTROUTER	IN=	OUT=br-73672de335db	SRC=172.18.0.7	DST=172.18.0.6	PROTO=TCP	SPT=45866	DPT=6379

SSL Report

Qualys. SSL Labs

Home Projects Qualys Free Trial Contact

You are here: [Home](#) > [Projects](#) > [SSL Server Test](#) > [ssl.itlab.ee.ncku.edu.tw](#)

SSL Report: ssl.itlab.ee.ncku.edu.tw (140.116.164.152)

Assessed on: Thu, 20 Jun 2019 01:17:57 UTC | HIDDEN | [Clear cache](#)

[Scan Another »](#)

Summary

Overall Rating **A+**

	Certificate	Protocol Support	Key Exchange	Cipher Strength
Overall Rating	100	100	100	100

Visit our [documentation page](#) for more information, configuration guides, and books. Known issues are documented [here](#).

HTTP Strict Transport Security (HSTS) with long duration deployed on this server. [MORE INFO »](#)

Qualys. SSL Labs

Home Projects Qualys Free Trial Contact

You are here: [Home](#) > [Projects](#) > [SSL Server Test](#) > [web.ncku.edu.tw](#)

SSL Report: web.ncku.edu.tw (140.116.241.66)

Assessed on: Thu, 20 Jun 2019 01:19:09 UTC | HIDDEN | [Clear cache](#)

[Scan Another »](#)

Summary

Overall Rating **A**

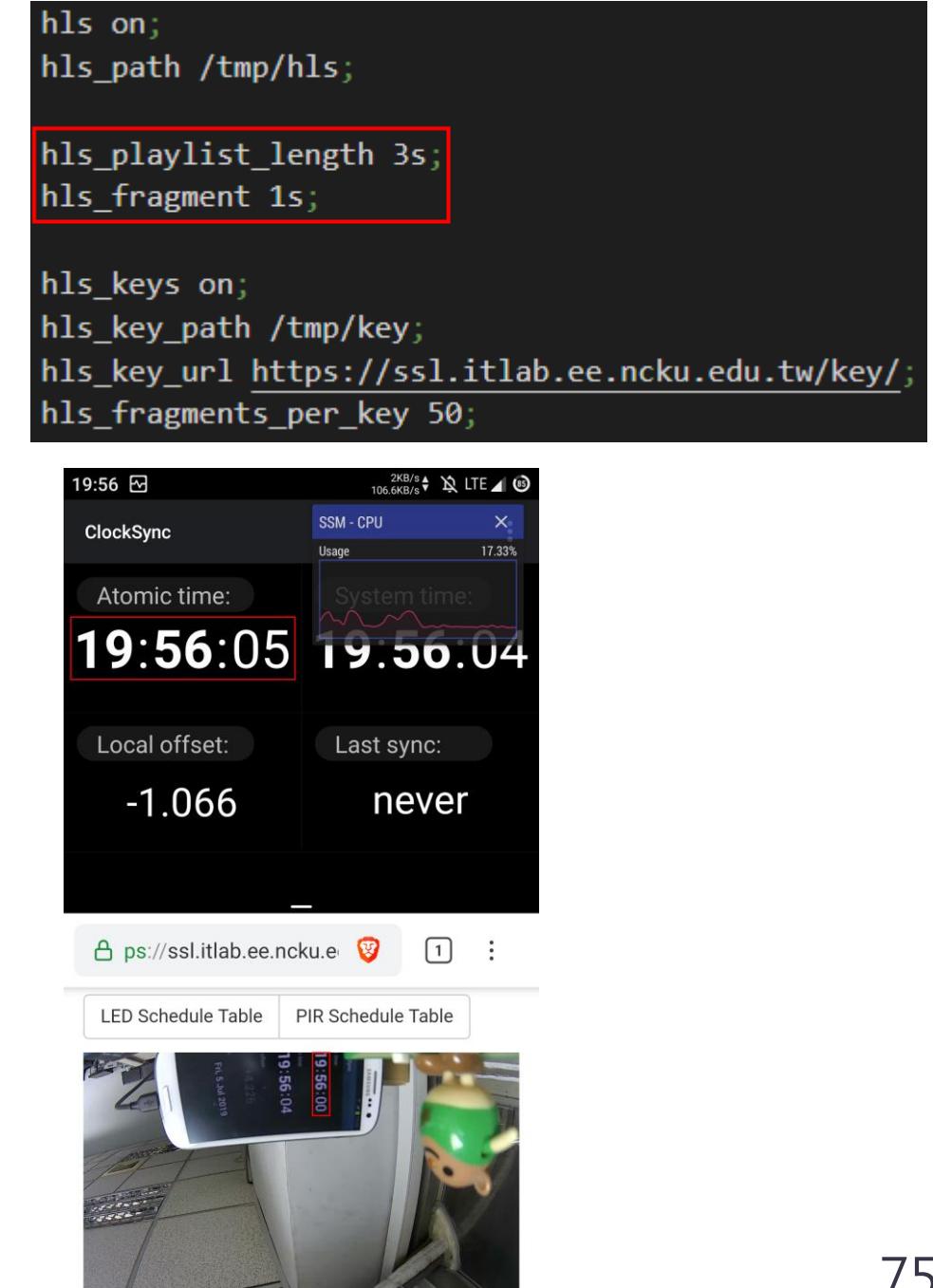
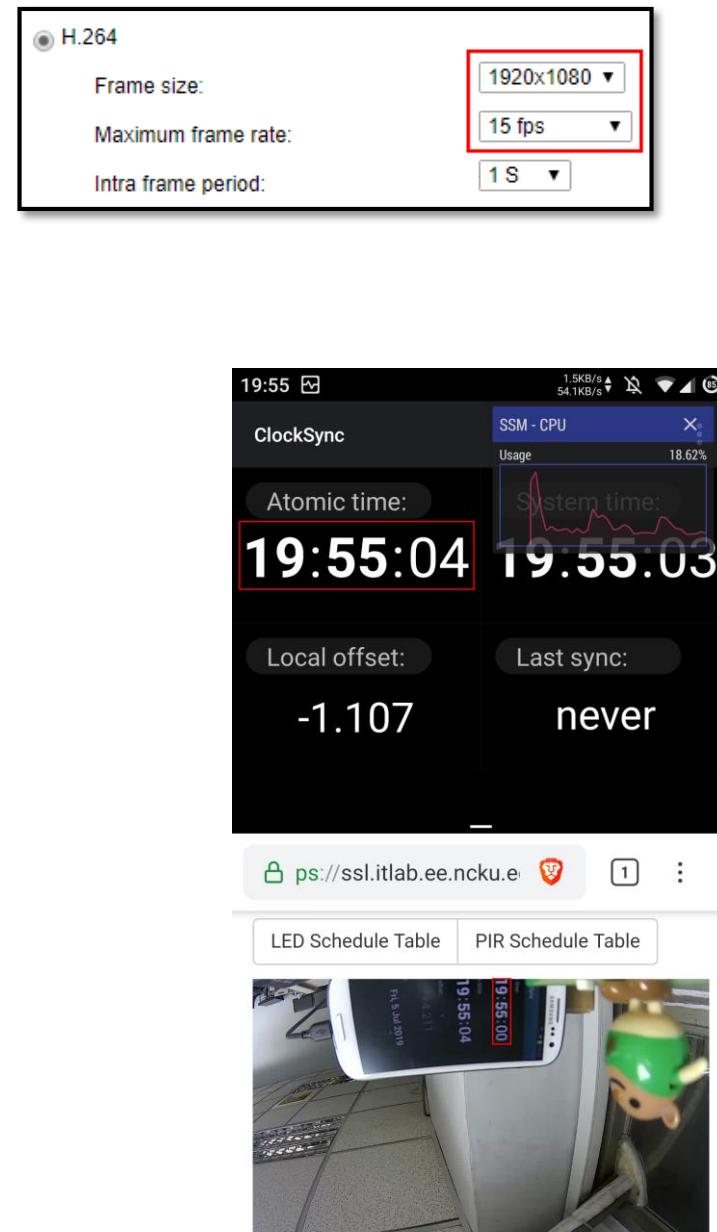
	Certificate	Protocol Support	Key Exchange	Cipher Strength
Overall Rating	100	95	90	90

Visit our [documentation page](#) for more information, configuration guides, and books. Known issues are documented [here](#).

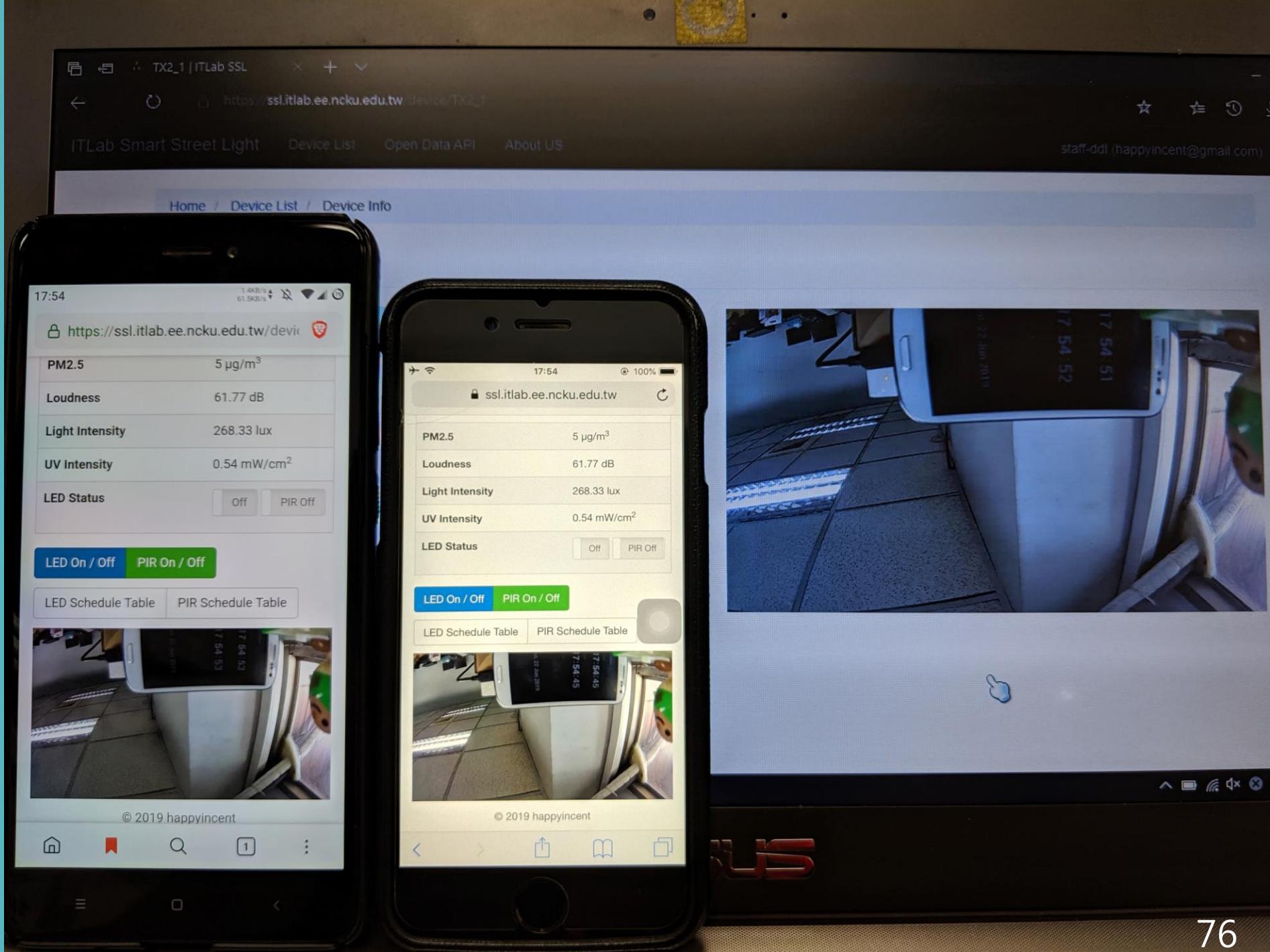
This server's certificate is not trusted by Java trust store (see below for details).

Video Latency in Live Streaming

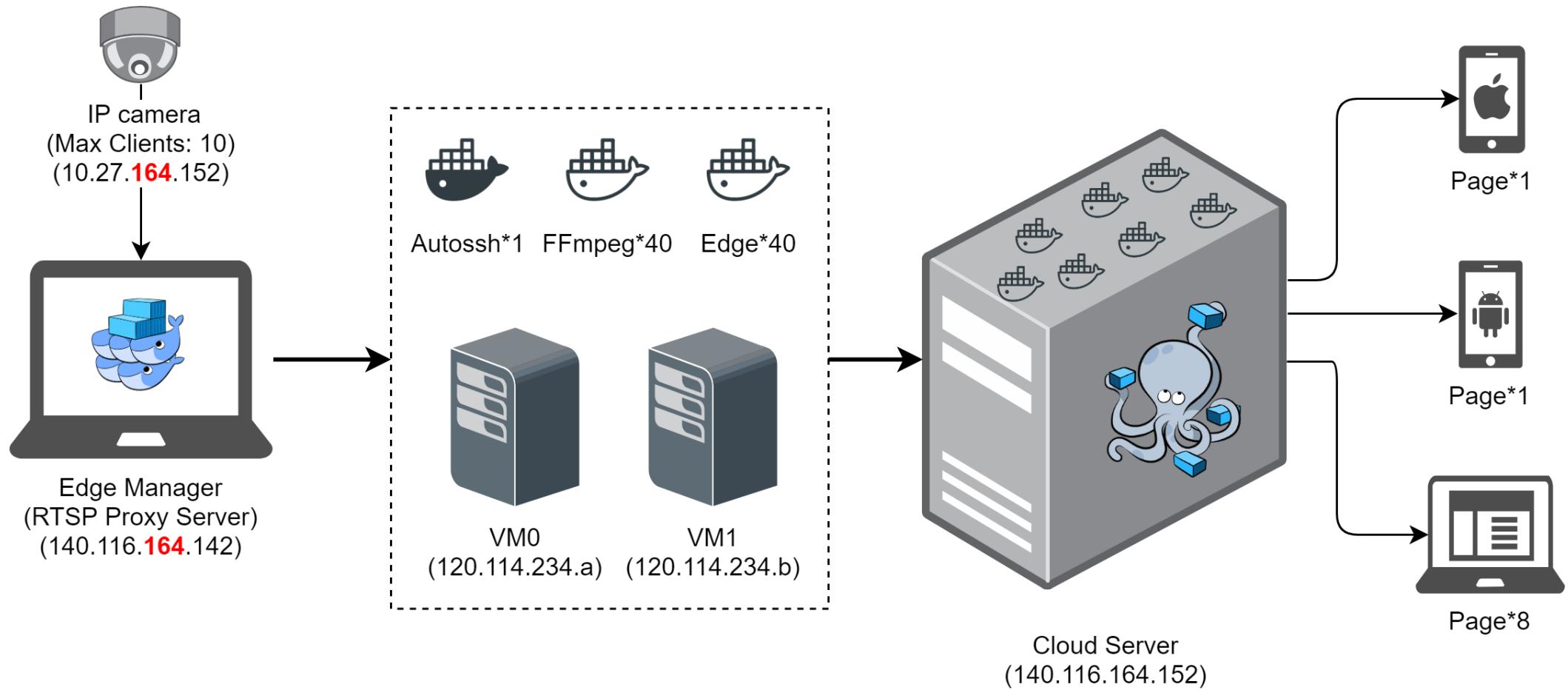
- (POE Switch in LAN)
- (Android 9.0)
- CPU ≈ 18 %
- Wi-Fi ≈ 4 s
- LTE ≈ 5s



Cross-Platform Compatibility



Performance of Cloud Server



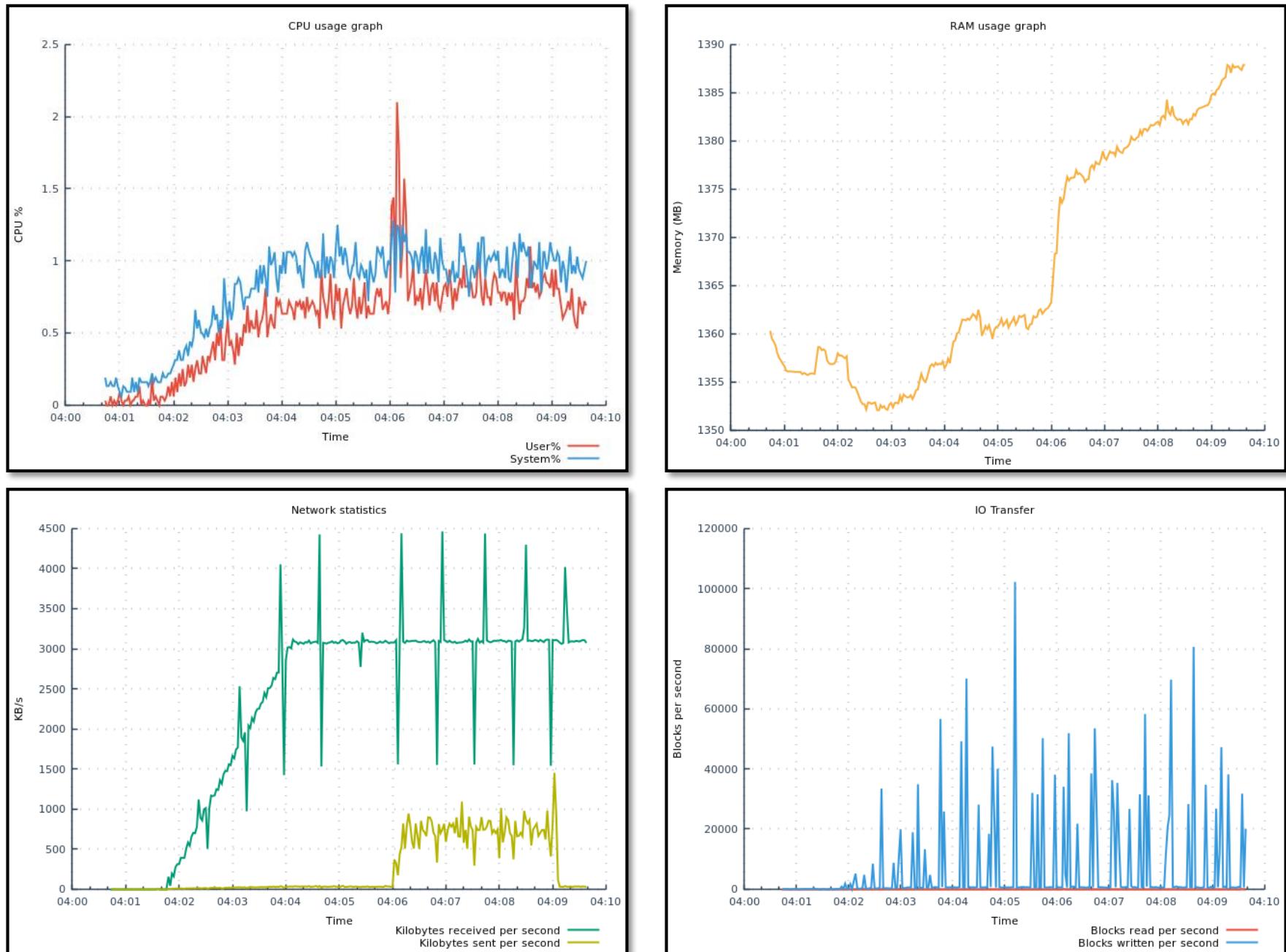
Performance of Cloud Server

04:02 ~ :

40 edge nodes

04:06 ~ 04:09:

10 browser clients

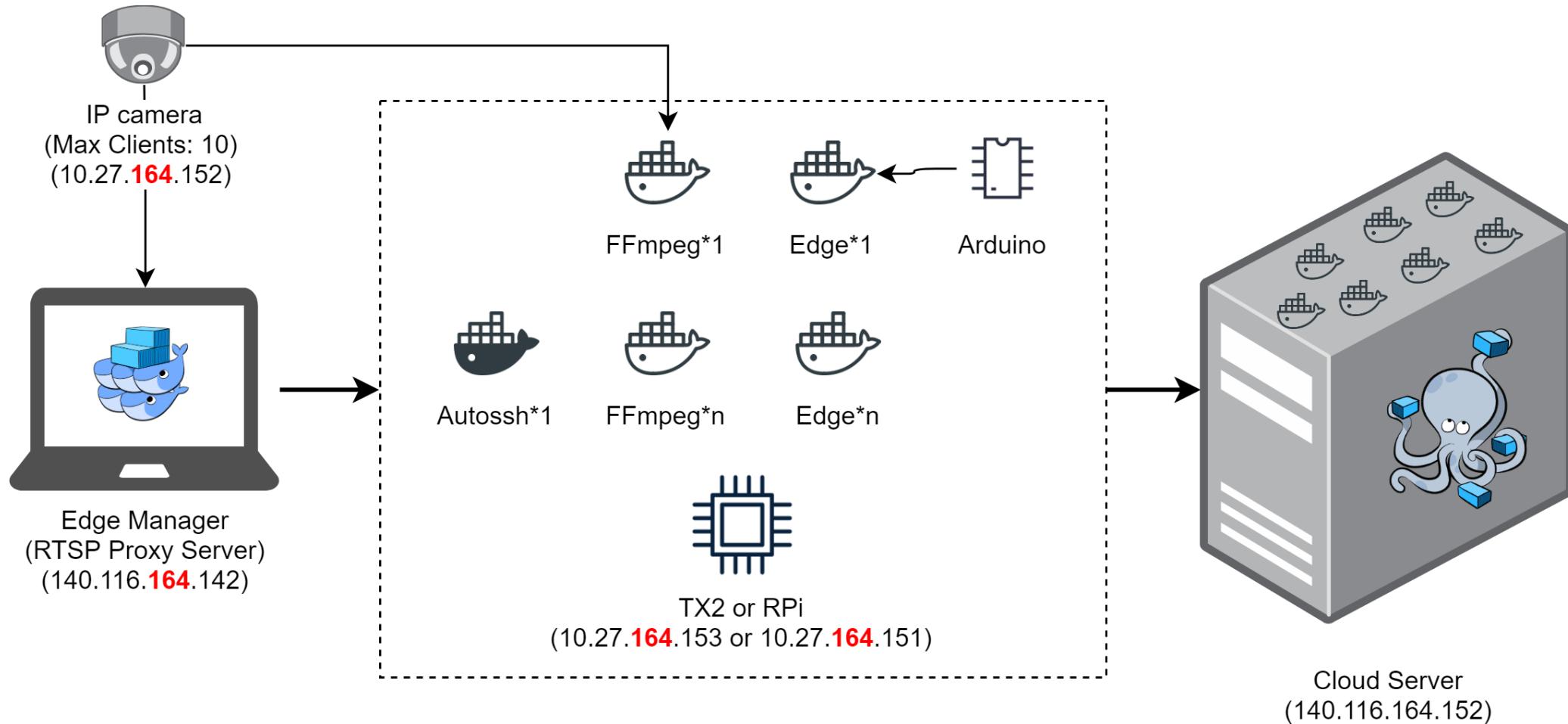


Performance of Cloud Server

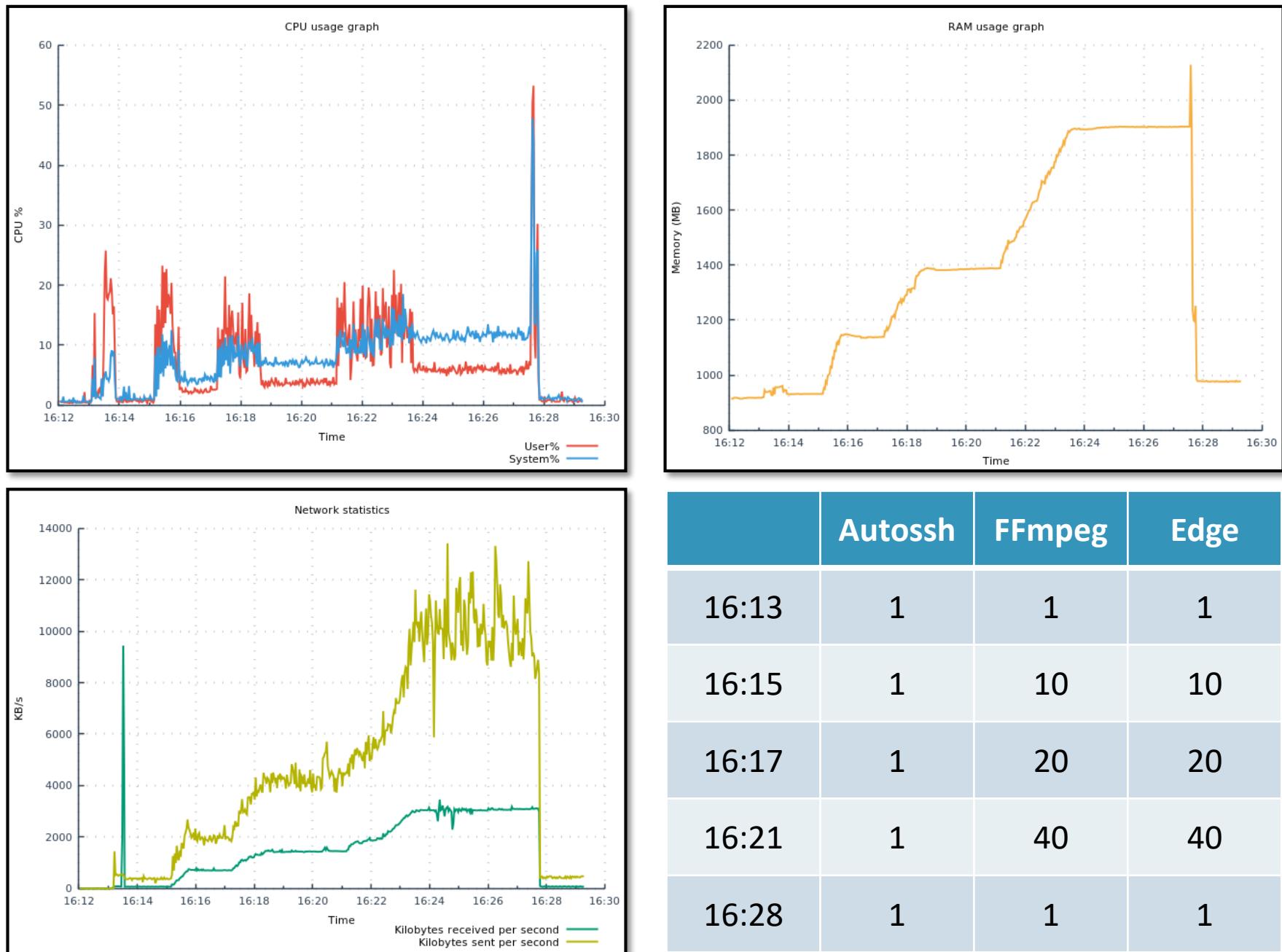
Table 4.4: CPU (%) and RAM (MB) usage of Cloud Server (average with 30 records)

	nginx	nginx-rtmp	certbot	web	mongodb	redis	sshd
0 edge, 0 client	0.00%	0.00%	0.00%	0.14%	0.32%	0.09%	0.03%
	28.67MB	8.04MB	3.67MB	261.60MB	41.88MB	3.46MB	23.27MB
1 edge, 0 client	0.00%	0.47%	0.00%	0.953%	0.56%	0.12%	0.30%
	28.88MB	8.15MB	3.67MB	280.58MB	41.90MB	3.47MB	24.31MB
1 edge, 1 client	0.17%	0.46%	0.00%	0.27%	0.37%	0.13%	0.31%
	28.88MB	8.17MB	3.67MB	262.62MB	41.96MB	3.48MB	24.29MB
1 edge, 10 clients	1.36%	2.32%	0.00%	0.24%	0.37%	0.11%	0.32%
	30.71MB	9.13MB	3.67MB	279.93MB	43.00MB	3.53MB	24.35MB
40 edges, 0 client	0.00%	12.04%	0.00%	1.40%	0.32%	0.38%	9.57%
	29.71MB	29.34MB	3.67MB	288.66MB	44.65MB	3.77MB	28.05MB
40 edges, 10 clients	1.48%	11.78%	0.00%	2.22%	0.37%	0.43%	9.37%
	30.64MB	43.42MB	3.67MB	289.64MB	44.97MB	3.89MB	28.33MB

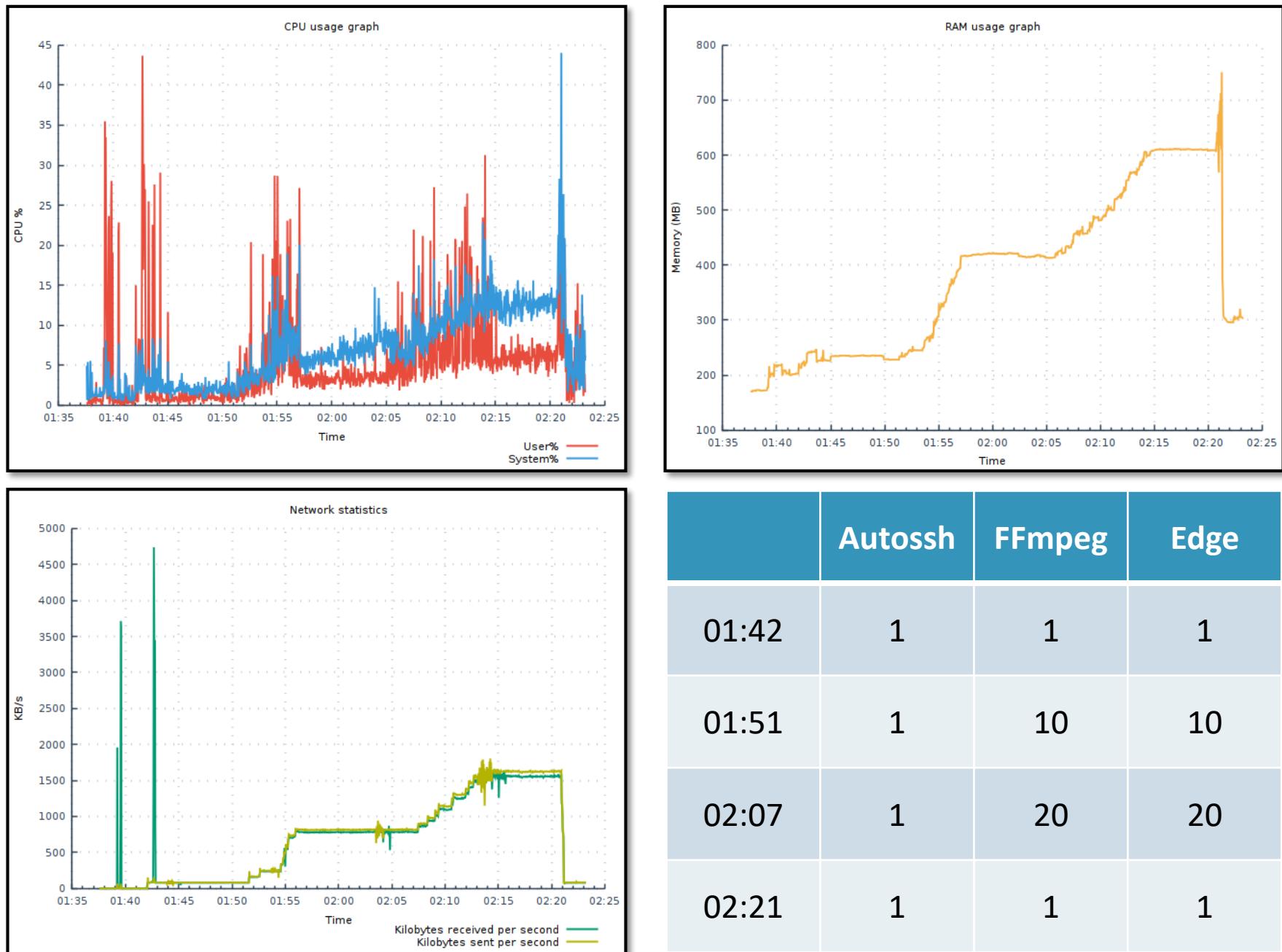
Performance of Edge Nodes



Performance of Edge Node (Nvidia TX2)



Performance of Edge Node (RPi 3B)



Outline

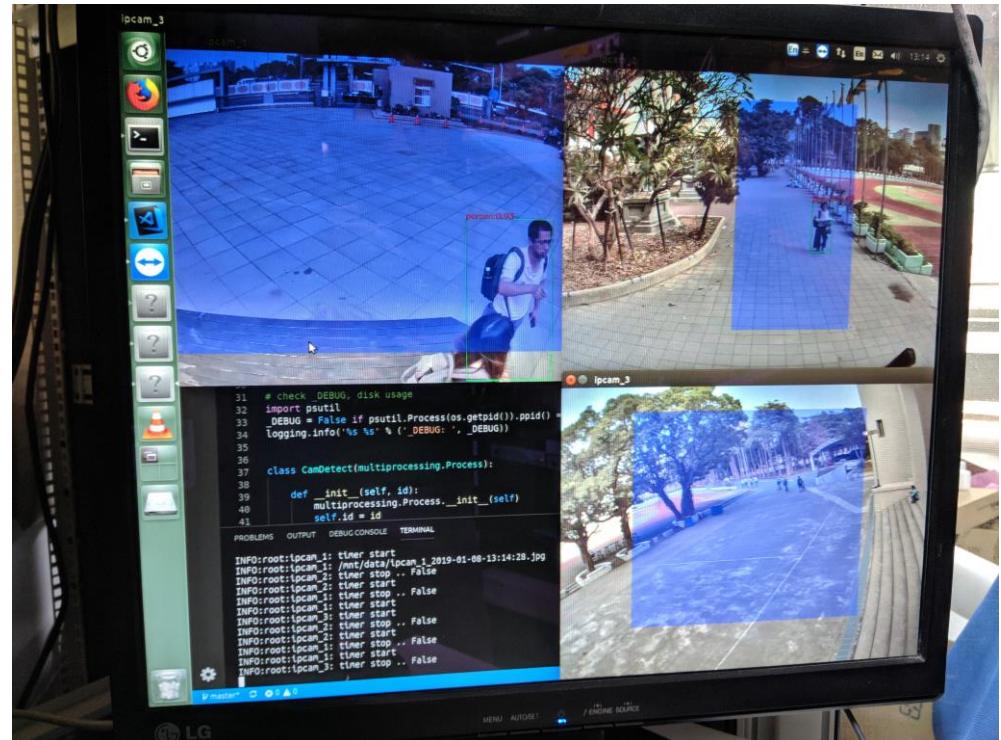
- Introduction
- Background and Related works
- System Design
- Experiment and Verification
- Conclusion and Future works

Conclusion

- Web-based streetlight management system
 - Lighting control (real-time)
 - Information of lamppost (real-time)
 - Historical information query API
- Secure Transmission (Cloud-Edge, Cloud-Browser)
- Containerize services in both Cloud and Edge

Future works

- Sensors management
- Intelligent applications
 - Smart lighting control mechanism
 - Real-time object detection (at Edge)
 - ...



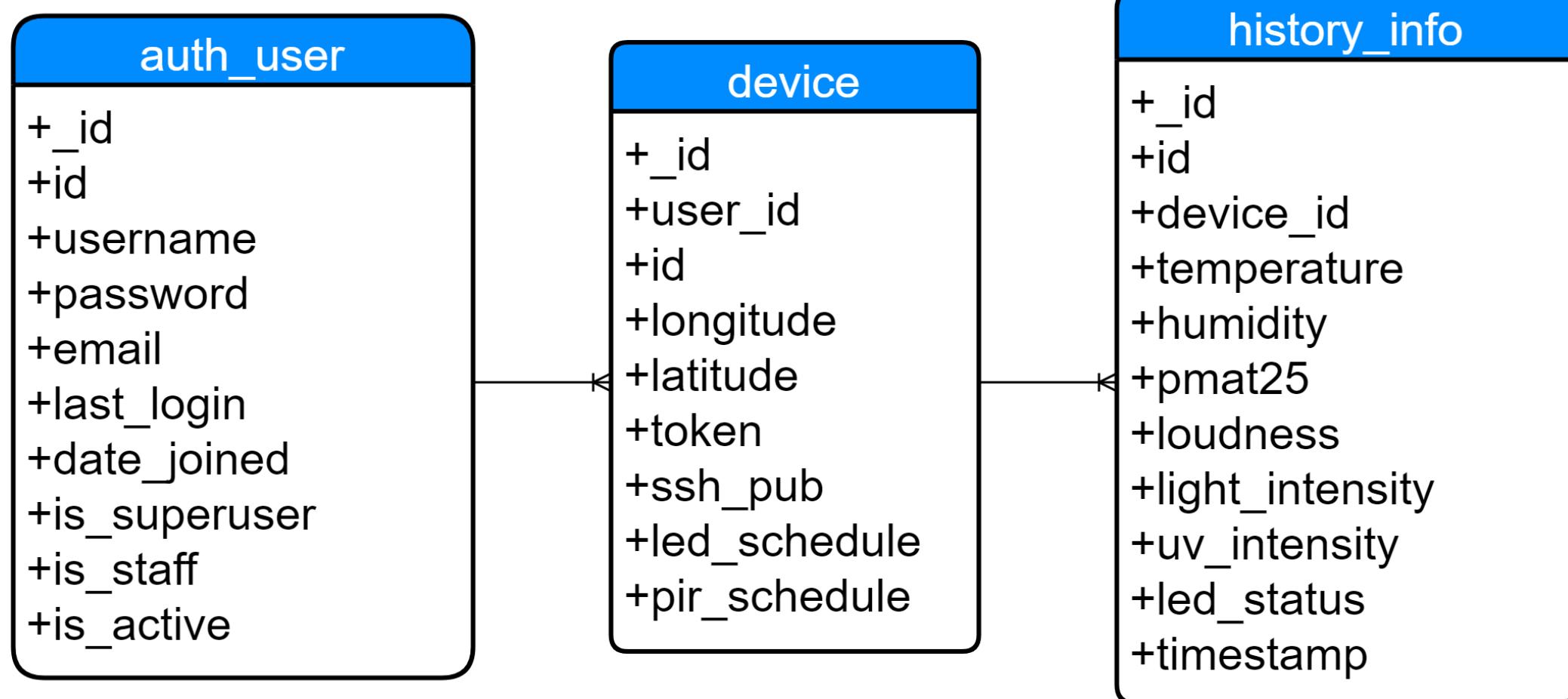


Thank you for listening !

Apendix

- Database (MongoDB)
- In-Memory Database (Redis)
- Edge.py
- Arduino.ino

Database (MongoDB)



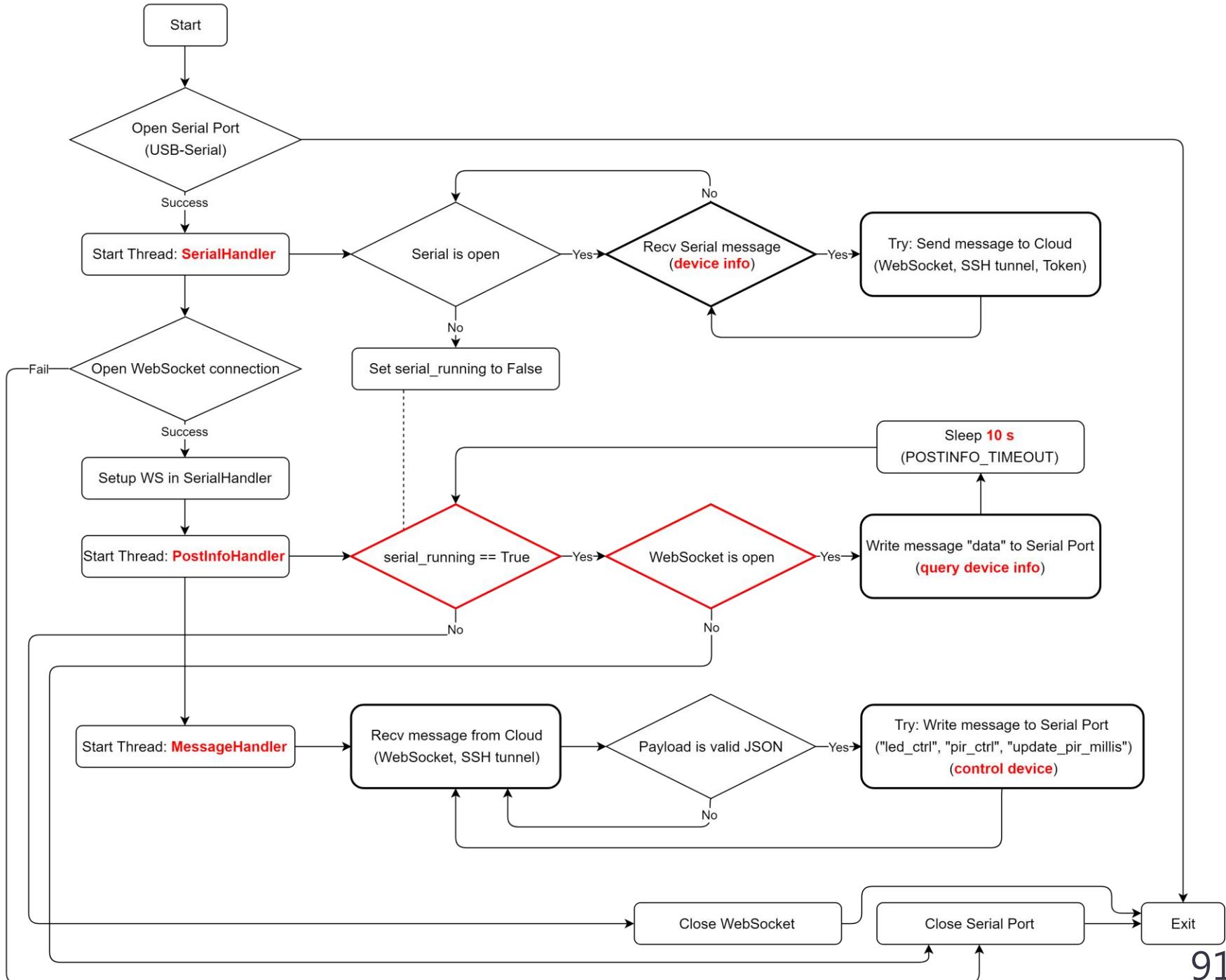
In-Memory Database (Redis)

Key	Value	TTL	Type
(session_key)	Session information	1 Day	Python pickle
(username)	Token for Open Data API	600 s	Python string
(Device ID)	Token for Edge services	-1 (∞)	Python string
(Device ID)_channel	WebSocket channel name of the Edge node	-1 (∞)	Python string
(Device ID)_info	Current information of the Edge node	30 s	Python pickle

Edge.py

(Threads)

- **SerialHandler**
(Arduino to Edge)
- **PostInfoHandler**
(Edge to Arduino)
- **MessageHandler**
(Cloud to Edge)



Arduino.ino

